



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A
ROBOTIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND
ROBOTICS

ANIMACE VÝROBNÍCH SYSTÉMŮ

PRODUCTION SYSTEM ANIMATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JOSEF VESELSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. SIMEON SIMEONOV, CSc.

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav výrobních strojů, systémů a robotiky

Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Josef Veselský

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Výrobní stroje, systémy a roboty (2301T041)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Animace výrobních systémů

v anglickém jazyce:

Production system animation

Stručná charakteristika problematiky úkolu:

Analýza problému.Návrh programového vybavení pro animaci výrobních systémů za využití OOP.Naprogramování DLL knihovny pro animace výrobních systémů

Cíle diplomové práce:

- 1.Analýza problému.
- 2.Návrh programového vybavení pro animaci výrobních systémů za využití OOP
- 3.Naprogramování DLL knihovny pro animace výrobních systémů

Seznam odborné literatury:

Brandimarte P., Villa A. „Modeling Manufacturing Systems“ ISBN 3-540-65500-X, Springer

Vedoucí diplomové práce: doc. Ing. Simeon Simeonov, CSc.


Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 20.1.2012

L.S.

doc. Ing. Petr Blecha, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

	Ústav výrobních strojů, systémů a robotiky	Str. 1
	DIPLOMOVÁ PRÁCE	

Abstrakt

Tato diplomová práce se zabývá vytvořením programového grafického rozhraní pro animaci výrobního procesu. Diplomová práce je rozložena na teoretickou a praktickou část. V teoretické části diplomové práci je uvedena rešerše vývoje software pro simulaci výrobního procesu. Dále jsou zde uvedeny základy používání simulace a animace, výhody a nevýhody. Teoretická část diplomové práce je ukončena popisem technologie a softwaru, který byl použit při vytváření grafického rozhraní pro animaci. Praktická část diplomové práce je zaměřena na vývoj programového rozhraní. Jsou zde popsány prvky, které byly použity při programování a vysvětleny základní funkce programu. Praktická část je ukončena shrnutím výstupních dat, které program uživateli poskytuje. K práci je přiloženo CD, na kterém je možno nalézt přímo zdrojový kód programu a výsledný hotový program.

Klíčová slova

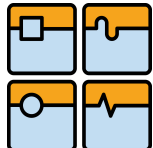
animace výrobního procesu, simulace, počítačová grafika

Abstract

This diploma thesis deals with the concept of creating a graphical user interface for the animation of a production process. The thesis is divided into theoretical and practical part. The theoretical part contains a research of the development of the software used for the process simulation. Further there are described the basics of using simulation and animation, its advantages and disadvantages. The theoretical part of this work is completed by the description of the technology and software, which was used for creating graphical interfaces for the animation. The practical part is focused on developing software interfaces. It describes the elements which were used in programming and there are explained basic functions of the software. The practical part is completed with a summary of the output data which the program offers the user. The thesis is accompanied by a CD, which contains the source code and the final vision of the program.

Keywords

animation production process, simulation, computer graphics

	Ústav výrobních strojů, systémů a robotiky	Str. 2
	DIPLOMOVÁ PRÁCE	

Bibliografická citace

VESELSKÝ, J. *Animace výrobních systémů*. BRNO: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. XXX s. Vedoucí diplomové práce doc. Ing. Simeon Simeonov, CSc.

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a použil jen prameny uvedené v seznamu použitých zdrojů.

V Brně dne 2.7.2012

_____ podpis

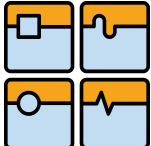
Poděkování

Touto cestou děkuji vedoucímu diplomové práce panu doc. Ing. Simeon Simeonovovi, CSc. za cenné rady, které mi v průběhu sestavování práce pomáhali. Také bych rád poděkoval všem, co mi pomáhali a podporovali mě po celou dobu studia.

OBSAH

1. Teoretická část.....	5
1.1 Úvod	5
1.2. Analýza problému	6
1.3. Historie simulačního software	6
1.4. Trendy v simulačním software	8
1.6. Počítačová simulace	9
1.6.1. Princip počítačové simulace výrobních systémů	10
1.6.2. Použití simulace	11
1.6.3. Výhody a nevýhody simulace	11
1.6.4. Oblasti využití simulačních softwaru	13
1.6.5. Sankey diagram.....	13
1.6.6. Ganttův diagram	15
1.7. Animace.....	16
1.7.1. Použití animace	17
1.7.2. Základy animace	17
1.7.3. Počítačová animace	18
1.8. Počítačová grafika	19
1.8.1. Rastrová grafika	20
1.8.2. Vektorová grafika.....	21
1.9. Software a technologie využité při vývoji	23
1.9.1. Microsoft Visual Studio .NET	23
1.9.2. .NET FRAMEWORK.....	24
1.9.3. ADO .NET	26
2. Praktická část.....	28
2.1. Popis a realizace vybavení pro animaci výrobních systémů.....	29
2.2.1. Realizace pracovního prostředí	32
2.2.2. Popis průběhu animace výrobním systémem	38
2.2. Výstupy simulace výrobního procesu	41
Zhodnocení.....	48
Seznam použitých zdrojů	49
Seznam použitých obrázků	50

Seznam použitých tabulek	51
Seznam použitých zkratk a symbolů	52
Přílohy	53

	Ústav výrobních strojů, systémů a robotiky	Str. 5
	DIPLOMOVÁ PRÁCE	

1. Teoretická část

1.1 Úvod

Jako téma své diplomové práce jsem si zvolil Animaci výrobních strojů. Jedná se o vytvoření grafického programového rozhraní, které by bylo schopno zobrazit chod výrobního procesu. Software pro simulaci a následnou animaci výrobních systémů je již delší dobu nedílnou součástí vědecké práce, ale i běžné inženýrské praxe. S neustálým rozvojem výpočetní techniky dochází také stále ke zdokonalování tohoto druhu softwarového vybavení. V dnešní době nejsou tyto programy zaměřeny pouze na řešení jednotlivých problémů, ale dovolují vyhodnocovat problematiku z několika různých pohledů. Při pohledu na množství oborů, ve kterých nalézá tento software využití, zjistíme, že dominují především obory technického charakteru od strojírenství přes elektrotechniku nebo chemii. V těchto odvětvích zpočátku sehrávaly svoji roli návrhové systémy typu CAD, do kterých postupem času začaly pronikat prostředky, které dovolovaly modelovat a simulovat chování navrhovaných prvků v nejrůznějších situacích.

Důvodů proč implementovat simulaci a modelování do výrobního systému podniku je velké množství. Mezi hlavní z nich bychom mohli řadit možnost simulovat ještě neexistující výrobu nebo u stávající výroby hledat příčiny vyskytujících se problémů a následně tyto problémy odstranit. Velice snadno lze získat tímto postupem přehled o úzkých místech ve výrobních procesech, vytížení jednotlivých strojů i personálu. Pomocí animačního softwaru můžeme také poskytovat prostředky pro zaškolování osob. V tomto případě může sloužit tedy ve funkci jako trenažér.

Další významnou výhodou tohoto software je časové hledisko. Je zřejmé, že simulaci výrobního systému můžeme provést v relativně krátké době. Tento čas se odvíjí od zkušeností uživatele a na komplexnosti modelovaného systému (čím přesnější model budeme chtít, tím více času se musí věnovat jeho návrhu). Také nesmíme zapomenout na ekonomickou stránku věci. V dnešní době se podnikům stále více vyplatí vynaložit náklady na pořízení takového programového vybavení, které dokáže poměrně přesně zobrazit chod podniku a případně odhalit nejrůznější úskalí, jež nejsou na první pohled zřejmá.

Nemusí se totiž vždy jednat o problémy technického či provozního charakteru, ale také o problémy ekonomické povahy. Proto mnoho dnešních simulačních programů mají v sobě implementovanou podporu ekonomických výpočtů a metod pro sledování kvality výroby. V dnešní době je možné získat celou řadu komerčních softwarů pro animaci výrobního systému pro platformy Windows i UNIX, které nabízejí nepřehledné množství možností.

1.2. Analýza problému

V této teoretické části diplomové práce je popsána historie simulačního softwaru, základy simulace, animace, počítačové grafiky a je zde zmínka o softwaru a technologii využitých při vývoji aplikace.

1.3. Historie simulačního software

První velmi jednoduché, ale i tak velice obtížně vypracované modely, se začaly objevovat kolem roku 1955. Parametry tehdejších počítačů byly pro využití simulačních úloh velmi nízké, rychlost provádění výpočetních operací byla u tehdejších výkonných počítačů řádově stovky aritmetických operací za vteřinu, operační paměti řádově tisíce až desítky tisíc slov, přídatné paměti byly velmi pomalé a grafika v té době prakticky neexistovala. K dispozici byly jen jednoduché programovací jazyky, například jazyky symbolických adres, první verze jazyka FORTRAN, později jazyk ALGOL. Samozřejmě s rozvojem výpočetní techniky se začala vytvářet speciální programová podpora, která sloužila pro řešení simulačních úloh. Ale tento software měl před sebou ještě opravdu dlouhou cestu vývoje k dosažení dnešní úrovně softwarových prostředků, kterých využíváme pro vytváření simulace. Ale ani současný stav tohoto software není zdaleka konečný a je v něm stále co zdokonalovat. [1]

Výkonnost simulačního software je pevně spjata s úrovní hardware, tj. na rychlosti počítačů, velikosti pamětí, síle grafických procesorů a dalších hardwarových komponent. Je proto samozřejmé, že rozvoj simulačního software pomalejším krokem sleduje rozvoj počítačového hardware. Dlouhodobý trend rozvoje počítačů byl predikován v minulosti panem Moorem, jedním ze zakladatelů společnosti Intel a již v roce 1963 předpovídal exponenciální charakter rozvoje počítačů. Podle tohoto tzv. Moorova zákona budou zdvojnásobovány technické parametry počítačů za každé dva roky. Charakter rozvoje simulačního software zhruba odpovídá dlouhodobé predikci pana Moora. [1]

Tyto simulační programy lze rozdělit do třech základních skupin:

Do první bychom mohli zařadit univerzální programovací jazyky jako FORTRAN, ALGOL, C, C++, atd. Tyto programovací jazyky nebyly navrženy přímo pro použití v simulaci, ale nacházejí zde uplatnění. Vlastnosti moderních programovacích jazyků, zejména objektově orientovaných, umožňují použít je relativně rychle pro naprogramování speciálních úloh. [1]

Do druhé skupiny patří programovací jazyky, jako např. SIMAN, SIMULA, GPSS atd. Tyto jazyky jsou strukturované, pracují s objekty a událostmi nebo procesy, umožňují porozumět základním konceptům a algoritmům, které probíhají při řešení simulačních úloh (i když jsou u vyspělých simulačních nástrojů často programátorovi a uživateli skryté). [1]

Třetí kategorie obsahuje celou řadu produktů, které jsou označovány jako simulační prostředí (někdy simulační systémy), které obsahují podporu mnoha aspektů při řešení simulačních úloh. Jejich součástí může být i některý ze simulačních jazyků z druhé kategorie, často mají modul označovaný jako analyzátor výstupů, dnes bývá k dispozici modul pro optimální řízení simulačních experimentů, běžný je modul podporující 2D a v poslední době zejména 3D animace. [1]

Simulační software se vyvíjel od roku 1955 do současnosti v tzv. šesti základních periodách:

- 1955-1960...perioda hledání
- 1961-1965...perioda nástupu
- 1966-1970...perioda formování
- 1971-1978...perioda rozvoje
- 1979-1986...perioda zdokonalování
- 1987- současnost...perioda integrovaných prostředí.

Perioda hledání

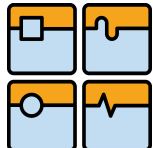
Pro řešení simulačních úloh se používaly hlavně obecné programovací jazyky. Úsilí bylo věnováno především hledání jednotících konceptů a vývoji procedur, které se mohly opakovaně používat pro různé simulační projekty. [1]

Perioda nástupu

Základy tehdejších simulačních programovacích jazyků tvořily jazyky FORTRAN, ALGOL a GPSS (General Purpose Simulation System). Nad FORTRANEM byly navrženy jazyky SIMSCRIPT a pozdější verze GASP (General Activity Simulation Program), ALGOL byl rozšířen o nové koncepty a vznikl jazyk SIMULA (SIMUlation LAnguage) a první verze GASP. Jazyky ALGOL a SIMULA vznikly a používaly se ve velké míře hlavně v Evropě, většina ostatních jazyků vznikla v USA. [1]

Perioda formování

V periodě formování byly na základě zkušeností a teoretických prací zdokonalovány předchozí verze simulačních jazyků a zejména díky rozvoji hardware začaly narůstat simulační aplikace (hlavní vliv na tom měl rozvoj počítačů IBM). Pro zajímavost jazyk SIMULA byl prvním jazykem, který zavedl jazykové komponenty jako třída, objekt nebo dědičnost a stal se tak předchůdcem, který inspiroval moderní objektově orientované jazyky. [1]

	Ústav výrobních strojů, systémů a robotiky	Str. 8
	DIPLOMOVÁ PRÁCE	

Perioda rozvoje

Simulační jazyky se začaly implementovat na další počítače a minipočítače. Posilovaly se softwarové prostředky pro efektivnější vývoj simulačních úloh (programové generátory, ladící programy), tedy hlavní úsilí bylo zaměřeno na zjednodušení návrhářské a programátorské činnosti. [1]

Perioda zdokonalování

V této periodě docházelo ke zlepšování stávajících programovacích jazyků (GASP, GPSS atd.), vznikly nové jazyky SLAM (Simulation Language for Alternative Modeling) a SIMAN (SIMulation ANalysis), které vytvořily nové pohledy na simulaci. Rozvinul se koncept plánování událostí, ve kterém se popis modelu tvořil pomocí blokových nebo jiných diagramů, bylo možno vytvářet kombinované modely obsahující spojitě i diskrétní prvky. [1]

Perioda integrovaných prostředí

Poslední perioda, která pokračuje až do současnosti, se vyznačuje bohatými implementacemi na PC a hlavně vznikem simulačního prostředí s grafickým uživatelským rozhraním, animačními a vizualizačními nástroji (od 2D k 3D zobrazování), analyzátorů vstupních a výstupních dat a použitím blokových diagramů a procesních přístupů ve fázi návrhu a realizace simulačních modelů. Výsledky různých simulačních běhů je možno tabulkově nebo graficky porovnávat a je také doplněno interaktivní řízení simulačních běhů. [1]

1.4. Trendy v simulačním software

Vysoká podobnost vzhledu modelů s realitou a přirozená, přehledná a hlavně srozumitelná práce se vstupními daty i výsledky je jednoznačným trendem v dlouhodobém vývoji simulačních prostředí dnešních programů. Tímto způsobem je v počítači tvořena virtuální realita, tedy funkční i vizuální iluze skutečného (i když často velmi zjednodušeného) reálného systému. Součástí tohoto pojetí v tvorbě modelů může být i možnost volby různé úrovně detailů použitých při experimentech a vizualizaci. [1]

Dalším trendem je standardizace rozhraní simulačních prostředků, které jsou využívány pro výměnu simulačních dat. V minulosti byl navržen standard SDX (Simulation Data Exchange), další práce se zaměřily na použití XML (eXtended Markup Language) standardů pro výměnu dat v simulaci včetně použití webu. [1]

Velké simulační modely je nutné simulovat na výkonných počítačích (serverech), které jsou prostřednictvím Internetu a vhodného počítačového rozhraní spojeny s klienty na lokálních počítačích. To znamená využití Internetu pro řešení simulačních úloh je dalším jednoznačným trendem v simulaci. [1]

Často se říká, že simulační modely je možné využít pouze pro předem jasné (a často velmi úzce) specifikované situace (hlavně z důvodu pracnosti modelů a výše nákladů). Toto staré simulační paradigma začíná být nahrazováno novým paradigmatem, podle kterého se modely tvoří ve více úrovních podle detailů, ale neformulují se všechny otázky, na které mají modely odpovědět (protože některé otázky zatím nebyly ani formulovány). Pokud má uživatel vytvořený model podle nového paradigma, bude schopen o to mnohem rychleji odpovědět na budoucí otázky. Samozřejmě takové modely jsou pracnější a nákladnější, ale rozvoj simulačního software toto podporuje a tedy usnadňuje. [1]

1.6. Počítačová simulace

Pro zvyšování konkurenceschopnosti podniku a při snaze o co nejefektivnější využití všech zdrojů nacházejí značné uplatnění metody, které se často shrnují pod názvem metody průmyslového inženýrství (racionalizační). Kromě těchto metod označovaných někdy také jako moderní metody (jako jsou „KANBAN“, „POKA-YOKE“, „KAIZEN“, „Totální údržba“ apod.) se stále větší důraz nabývá i na využívání počítačové simulace diskrétních událostí. To je způsobeno jejími vlastnostmi, které umožňují poměrně rychle vyzkoušet a posoudit různé varianty řešení a tím minimalizovat rizika chybných rozhodnutí. Důležitější roli hraje také fakt, že analytické metody jsou pro řešení dnešních komplexních, dynamických úloh příliš statické a často narážejí na hranice svých možností. Také vývoj moderních simulačních systémů usnadňuje a urychluje zpracování simulačního projektu takovým způsobem, že dovoluje jejich takřka každodenní použití. Počítačová simulace diskrétních událostí je tedy v dnešní době nezbytným podpůrným nástrojem v oblasti zefektivňování provozu průmyslových výrobních systémů. [2]

Přes mohutný nástup počítačových aplikací a vývoj moderních simulačních systémů je počítačová simulace diskrétních událostí stále tuzemskou praxí nedoceňována, i když v zahraničí se již stala nedílnou součástí každodenní činnosti. Jejímu většímu rozšíření zabraňuje nejen poměrně vysoká cena simulačního softwaru, ale také hlavně nízká úroveň znalostí pro její plnohodnotnou aplikaci.

V dnešním výrobním prostředí již nelze dosáhnout efektivního provozu systémů pouze lokální optimalizací jednotlivých parametrů nebo subsystémů. Ty se totiž natolik vzájemně ovlivňují, že není možné řešit tyto parametry nebo subsystémy samostatně, nýbrž je třeba na celý systém nahlížet globálně a hledat optimum systému jako celku. Vysoká dynamika, komplexnost a tím i nejistota při řešení podnikových úloh stále více vyžadují využívání takových metod a podpůrných prostředků, které umožňují komplexní přístup k projektování výrobních systémů i rychlé vyzkoušení různých variant řešení. Významným pomocníkem se proto v dnešní době stále více stává komplexní dynamická analýza výrobních systémů pomocí počítačové simulace. [2]

1.6.1. Princip počítačové simulace výrobních systémů

Počítačovou simulaci výrobního systému můžeme chápat jako etapu dynamického zkoumání výrobního systému. Jejím hlavním principem je experimentování s počítačovým modelem, který je pokud možno co nejpřesnějším obrazem stochasticky se chovajícího výrobního systému. Na simulačním modelu se mohou simulovat stavy systému (transport součástí, proces obrábění, porucha) v závislosti na uplynutém čase. Jejich změna nastává diskrétně v časových okamžicích, které mohou být reprezentovány diskrétní událostí (příchod zakázky, začátek obrábění, začátek seřizování, začátek poruchy apod.). Požadovaným výsledkem experimentování může být vyhledání hodnot výstupních veličin modelu, které vyhovují předem stanoveným požadavkům (cílům simulační studie). Vstupní hodnoty tohoto řešení jsou následně využitelné také pro reálný systém. [2]. Pro upřesnění je princip simulace výrobního systému znázorněn na obr.1 .



Obr. 1 Princip simulace [2]

Použití simulace je vhodné pro: [1]

- rozhodování o investicích
- prověřování vlivu zamýšlených inovací
- hledání a prověřování vhodných pravidel řízení
- posuzování a nalézání vhodné sortimentní struktury výrobků, prověřování splnitelnosti výrobních plánů
- zvýšení průchodnosti (např. obrobků) výrobním systémem s minimálním stavem zásob a minimálními provozními náklady
- modelování informačních toků a sítí
- plánování reorganizací, odstávek, údržby, výměn agregátů a využití pracovních sil
- hledání režimů vyhovujících ekonomickým a ekologickým omezením
- spolehlivostní analýzu složitých systémů
- důkladné poznání složitých systémů

1.6.2. Použití simulace

Simulaci je vhodné použít, když: [1]

- zkoumaný systém je natolik složitý, že neexistuje vhodná matematická metoda na řešení úlohy - to je vhodné zejména u komplexních systémů (výrobní podniky, servisní organizace, klastry podniků, atd.)
- je třeba posoudit operační parametry i mimořádné situace a reakce na ně nově projektovaného systému
- je třeba ověřit analytická řešení
- zkoumaný systém mění své vlastnosti příliš pomalu
- zkoumaný systém mění své vlastnosti příliš rychle
- zkoumaný systém by při špatně voleném experimentu způsobil katastrofu sám sobě nebo svému okolí, přičemž nebezpečí experimentu není předem známo
- ekonomický systém je těžko manipulovatelný (nelze dělat pokusy, pokusy jsou nákladné, atd.)
- je vhodné ji použít pro trénink (tzv. simulátory - trenažery)
- je vhodné vizualizovat chování systému pomocí animace spojené s průběhy simulačních experimentů - vytvoření vizualizace virtuální reality

Simulaci není vhodné použít, když: [1]

- řešení problému je možné získat analyticky
- je snazší a lacinější provést experimenty na reálném systému
- náklady na simulaci převýší předpokládané úspory
- náklady na simulaci překročí dostupné finanční prostředky zadavatele
- doba potřebná na získání výsledků ze simulace přesáhne potřebné termíny zadavatele
- nejsou k dispozici data potřebná pro provedení simulačních experimentů
- nebude možno dostatečně validovat model (z důvodu času, nedostatečné podpory expertů na problematiku od zadavatele, atd.)
- manažeři zadavatele mají nerealistická očekávání („příliš brzo, příliš mnoho“)
- složitost reality je tak velká nebo chování natolik neurčité, že nemůžou být dosti přesně definovány

1.6.3. Výhody a nevýhody simulace

Na rozdíl od optimalizačních modelů, simulační modely spíše zobrazují, než nabízejí určité řešení. Simulační model pracuje s danou množinou vstupů a vlastností a výstupy z modelu se pozorují a vyhodnocují. Pro nové výběry z uvedených množin se musí získat nové výstupy, které je třeba opět znovu vyhodnotit. Vyhodnocovací nástroje tedy negenerují nová optimální řešení, ale pouze vyhodnocují zadané parametry z hlediska jejich chování. Nejideálnější možnosti, které jsou získány z množiny řešení, jsou potom doporučena k implementaci. [1]

Mezi výhody simulace patří: [1]

- nová řešení, modifikace starých, rozhodnutí o nejvhodnější variantě atd., to vše může být zkoumáno bez přerušení probíhajících operací v reálném systému
- lze zkoumat vnitřní vazby mezi veličinami modelovaného systému
- lze porozumět vlivu důležitých veličin na výkonnost systému
- lze provést analýzu úzkých míst v reálných systémech
- simulační studie může pomoci zjistit, jak reálný systém skutečně funguje, a ne jak si jednotlivci myslí, že funguje
- odpovědi na otázky „co se stane když...?“ zejména u nově projektovaných systémů
- možnost komplexně zachytit dynamické a stochastické vazby v systému
- možnost experimentovat tak, jak by to v reálném světě z různých důvodů nešlo (nákladnost, nebezpečnost, zdlouhavost, neexistence systému - projekt)
- možnost pracovat v kvalitním programovacím jazyce nebo simulačním systému

Mezi nevýhody simulace patří: [1]

- tvorba modelů vyžaduje speciální trénink a znalosti, tvorba je tak trochu „uměním“ získaným zkušeností a časem
- u některých simulačních úloh je obtížné interpretovat výsledky, protože výstupy jsou v podstatě náhodné veličiny vzhledem k náhodným vstupům; proto může být někdy obtížné rozlišit, zda některá pozorování jsou důsledkem vnitřních vztahů v systému nebo důsledkem náhody v simulačních experimentech
- řešení simulační úlohy může být časově i nákladově náročné; snaha zadavatele po lacinějším řešení může pak vést k nekvalitním analýzám a modelům, které vyústí v nesprávné výsledky
- simulační řešení jsou někdy požadována pro úlohy, které mají analytické řešení (např. dále uvedený model myčky aut, který je analyticky řešitelný jako úloha teorie front - zde se uvádí jen jako ukázka tvorby modelu); taková simulační řešení je nutno důkladně zvážit a dohodnout se zadavatelem

Pro vykompenzování některých z uvedených nevýhod simulace lze uvést: [1]

- prodejci simulačního software vytváří problémově orientované softwarové balíky, které obsahují modely nebo jejich části pro přímé použití (zadají se jen parametry a data), pro takové obecně použitelné modely se užívá označení template
- mnohé simulační systémy mají vyvinuty analyzátory výsledků, které provádí důkladnou analýzu výstupních dat
- simulace se provádí dnes rychleji než včera, zítra se bude provádět rychleji než dnes

1.6.4. Oblasti využití simulačních softwaru

Některé oblasti, kde se simulační software používá: [1]

- výrobní systémy
- dopravní systémy
- logistické úlohy
- distribuční aplikace
- vojenské aplikace
- řízení lidských zdrojů
- modelování a řízení podnikových procesů
- atd.

Možné výstupy simulačního software např. od modelu výrobního procesu: [1]

- velikosti průběžných dob výroby součástí
- využití jednotlivých pracovišť
- analýzu prostojů pracovišť
- čekání na palety
- čekání na dopravník palet
- čekání na nástroje
- charakteristiky front palet
- atd.

Některé výstupy, které jsou součástí praktické části práce, jsou i blíže popsány v této teoretické části. Jsou zde uvedeny základní informace k Sankeho a Ganttovým diagramům. Tyto diagramy jsou jedny z výstupních informací, kterými jsme schopni nahlédnout na průběh výrobního systému.

1.6.5. Sankey diagram

Sankey diagramy získaly jméno podle irského inženýra Matthew Henry Phineas Riall Sankey, který jako první použil tento typ diagramu v článku pro energetickou účinnost parního stroje v roce 1898. Sankey diagram je obecné grafické znázornění určité veličiny toků, jako mohou být energie, materiály nebo peněžní toky oddělující se od základny, nebo náklady na převody mezi procesy. Sankey diagramy se v dnešní době používají pro zobrazování průtoku množství technologických systémů. Používají se nejen ve fyzice a technice, aby demonstrovaly, jak je energie distribuována, ale i například pro cash flow ve firmách. Mohou být užitečné při hledání hlavních příspěvků k celkovému toku. Graficky znázorňují tok materiálu mezi jednotlivými pracovišti na základě půdorysného plánu objektu.

Pro grafické znázornění je možné použít maticovou tabulku, která udává přepočtené množství přepravovaného materiálu mezi jednotlivými pracovišti ve zvolených jednotkách. Často jsou toky mezi jednotlivými pracovišti znázorněny ve formě šipky. Šířka šipek udává velikost úměrnou představovanému toku vzhledem k celku. Takto zjištěné množství materiálu je v Sankey diagramu znázorněno šířkou plných šipek, které současně označují směr toku materiálu. Pro větší názornost bývají jednotlivé šipky rozlišeny barevně. Šířka šipek je v poměru k průtoku množství. Sankey diagramy jsou však specifickým typem schématu. Sankey diagram poskytuje vizuální znázornění vstupních nebo výstupních situací.

Sankey diagramy umožňují vizualizovat proudění, proces nebo systém mnohem snadněji než zpracované tabulky s číselnými údaji.

Příklady, kde můžeme využít Sankey diagramy:

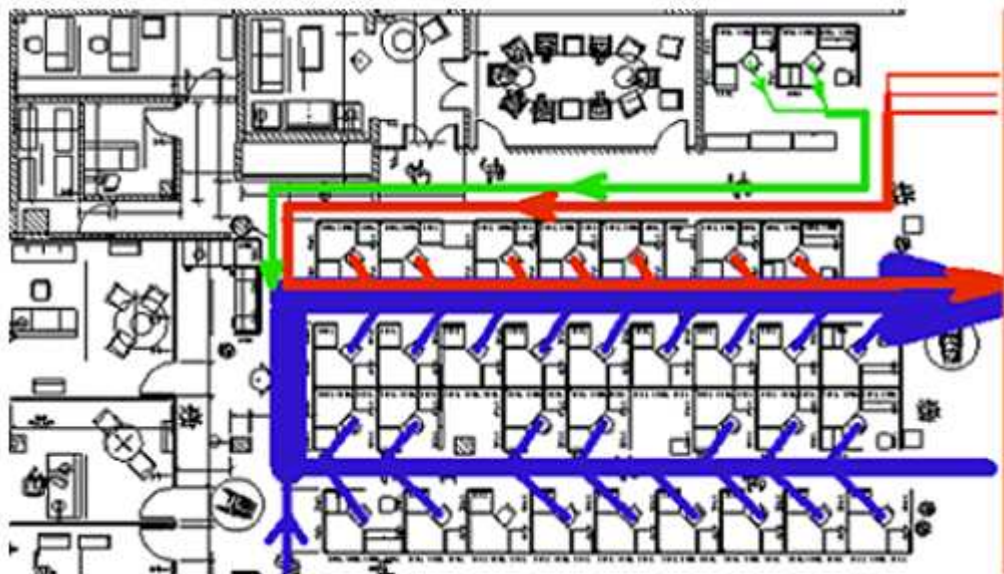
- dopravního proudu ve městě
- tok energie ve výrobním závodě
- průtok vody přes síť nádob a potrubí
- tok peněz přes národní ekonomiky

Pro navrhování dispozičního řešení pracovišť ve výrobním systému, v rámci vylepšení materiálového toku, je využíván také Sankey diagram, který graficky zachycuje intenzitu toku materiálu. Předpokladem pro jeho sestavení je zohlednění veškerého materiálového toku, který výrobním systémem protéká. Pro tento diagram je charakteristické to, že šipka definuje směr toku, šíře šipky odpovídá s objemem přepravovaného množství a délka vypovídá o přepravní vzdálenosti mezi jednotlivými pracovišti. Existují i typy Sankey diagramů, které nezachycují pouze materiálové toky, nýbrž i toky informací, pracovníků, energií atd. Za pomoci tohoto diagramu lze vyjádřit i tok pracovníků například v případě, když se zpracovává návrh příchodu a odchodu pracovníků ze závodu.

Programové nástroje, které pracují se Sankey diagramy, můžeme rozdělit do tří základních skupin:

- programy, které jsou speciálně vytvořeny pro tvorbu různých Sankey diagramů
- programy, nástroje, které podporují jeden konkrétní typ Sankey diagramu
- programy, které mají využití Sankey diagramu jako jednu ze svých funkcí, ale jejich hlavní účel je jiný

Příklad programu, který využívá Sankey diagram jako jednu ze svých funkcí, je znázorněn na obr. 2. Jedná se o jeden z možných výstupů simulačního programu Plant Simulation.



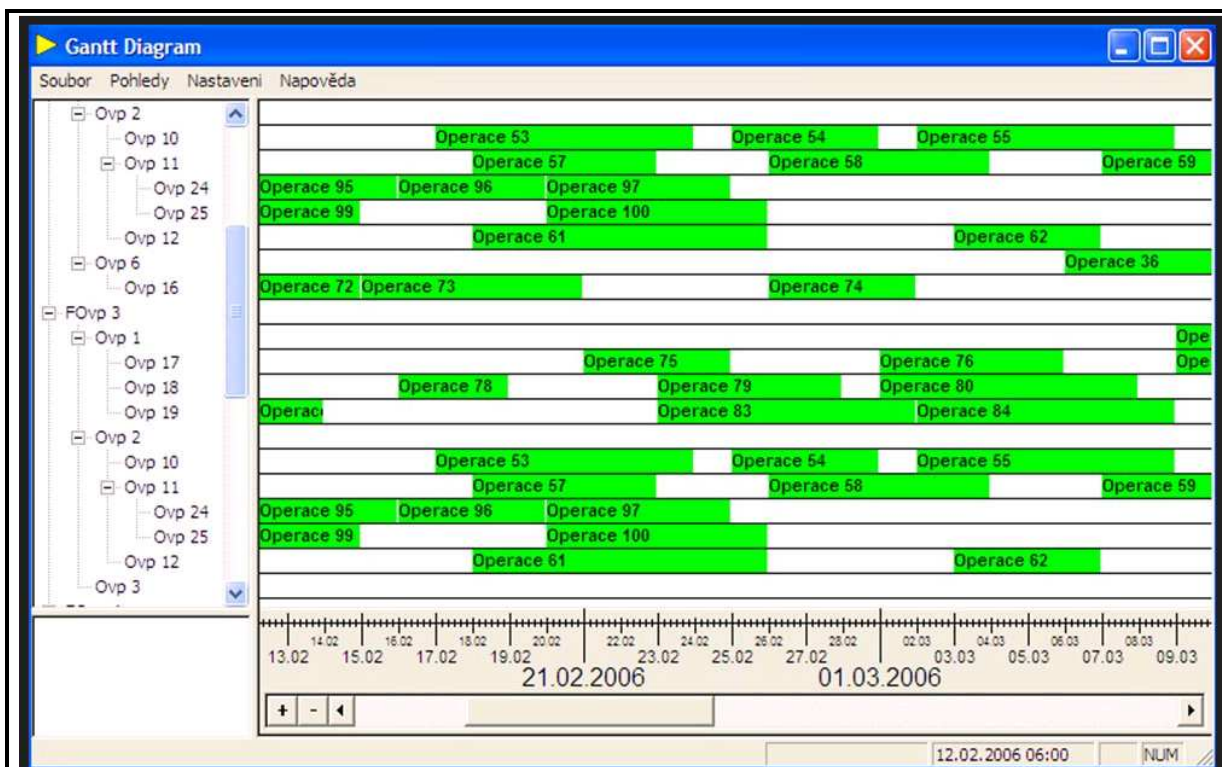
Obr.2 Příklad použití Sankey diagramu v programu Plant Simulation [3]

1.6.6. Ganttův diagram

Tento diagram byl poprvé použit Henrym L. Ganttem na základě analýzy pracovních postupů v průmyslové výrobě. Při vývoji moderních nástrojů pro lineární plánování a řízení projektu došlo k účelnému rozšíření tohoto původně lineárního diagramu zejména pro různé prezentace síťových grafů a hierarchických datových struktur.

Ganttův diagram umožňuje přehledně prezentovat aktuální stav projektu, směrný a aktuální plán, zejména údaje časového rozvrhu, práce, nákladů, financování a zisku na projektu. K aktualizaci a prezentaci závislostí mezi úkoly poskytuje Ganttův diagram strukturu na časové stupnici, zejména ke znázornění důležitých termínů. Prezentace souhrnných úkolů mohou poskytovat požadované sumarizace hodnot směrných, aktuálních a současných plánovaných ukazatelů podle reálné situace projektu

V současné době Ganttův diagram patří k nejpoužívanějším formám prezentace projektových modelů pro plánování a řízení rozsáhlých projektů. Hlavní výhodou Ganttova diagramu je tedy přehlednost projektových ukazatelů na časové ose a přehled hierarchické struktury projektu. Náznorný příklad Ganttova diagramu můžeme nalez na obr.3.



Obr.3 Příklad použití Ganttova diagramu v aplikaci ASP Karat [4]

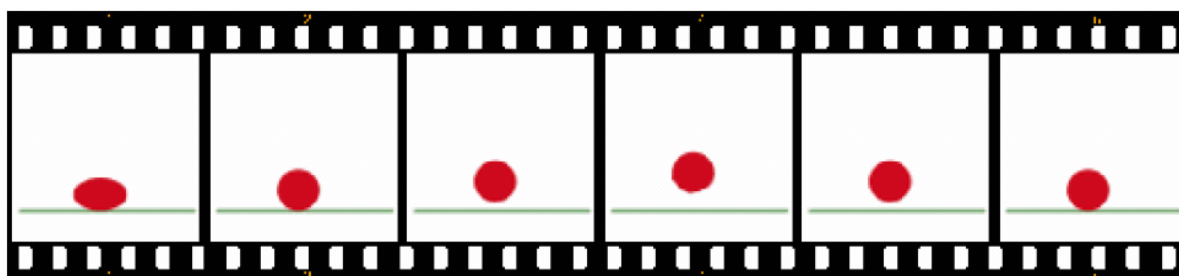
1.7. Animace

Pod slovem animace si můžeme představit proces, při kterém vzniká zdánlivá iluze pohybu za použití nepohyblivých objektů. Samotný pojem animace původně pochází s cizího slova znamenajícího oživení.

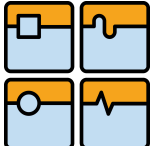
Díky nedokonalosti lidského oka, která se nazývá setrvačnost lidského oka, mohla vzniknout animace. Tato nedokonalost zajišťuje uchování obrazu po určitou dobu po přijetí na sítnici. Díky tomu lze vytvořit zdánlivou iluzi pohybu rychle se střídajícími obrázky.

Základem animace je tedy zaznamenávání sekvencí nepohyblivých snímků, které se od sebe liší např. pozicí a následně tyto snímky zobrazovat takovou rychlostí, při které se uplatní setrvačnost lidského oka a vznikne iluze pohybu.

Klasický film využívá rychlost 24 snímků za sekundu. Při této rychlosti již lidské oko pozoruje spojitý obraz. Náznorný příklad je na Obr. 4. Pokud bychom tuto sekvenci pustili rychle za sebou, vznikl by dojem skákajícího balonku. [5]



Obr. 4. Příklad průběhu animace [5]

	Ústav výrobních strojů, systémů a robotiky	Str. 17
	DIPLOMOVÁ PRÁCE	

1.7.1. Použití animace

V technických odvětvích slouží animace jako prostředek pro zobrazení reálných procesů. Zde je možné animovat děje, které jsou časově náročné, nákladné a nebezpečné při reálném provedení. Lze simulovat různé kritické situace a animovat různé situace, které ve skutečnosti nelze z různých důvodů vůbec provádět. [10]

Využití animace ve výuce usnadňuje studentům snáze pochopit probíranou látku. Lze popisovaný proces zobrazit a vysvětlovaný děj znázornit o to přehledněji. A pokud je výuková látka doplněna krátkou ukázkou a student si tuto může opakovaně pustit a vracet zpětných chodem, jde o význačnou studijní pomůcku pro studenty. [10]

Další mohutné využití animace je spojeno s filmem. Asi nejznámější studio spojené s pojmem animace je studio Pixar (1984), které pod vedením Jana Pinkavy vytvořilo celou řadu animovaných filmů. Jako jeden z nejznámějších se uvádí Gerl's Game. V roce 2007 získal snímek Oscara. Jde o zcela animované filmy. Animace ve filmové tvorbě může však být využívána i jako doplněk filmu. Jedná se zde o náročné scény, které jsou nebezpečné a mnohdy nerealizovatelné. [10]

V architektuře nachází animace také uplatnění. Možnost simulování situace, která ještě nenastala (namísto konstrukce pevných - nákladných modelů). Jde o modelování vzrůstu vegetace za určitou dobu. Zde je využití animace doplněno o další disciplíny, jako je chemické a fyzikální procesy, modelování vzrůstu stromů a pod. [10]

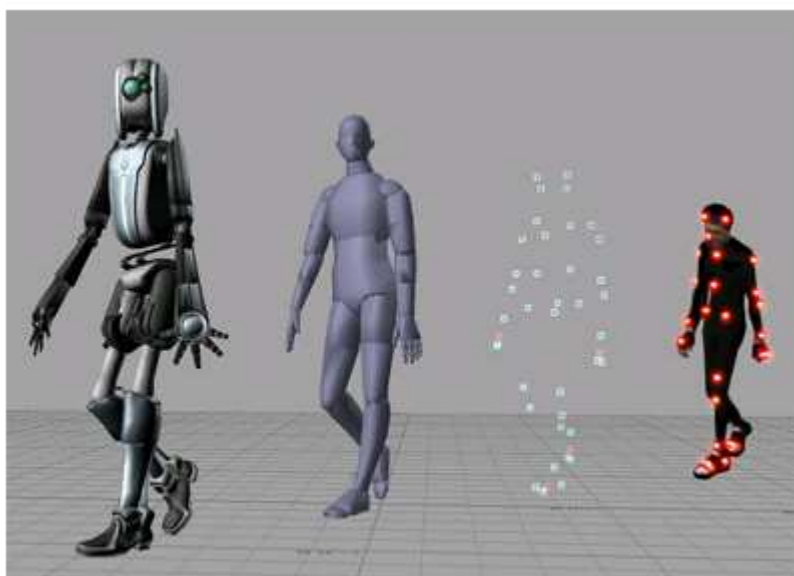
1.7.2. Základy animace

Základem animačních algoritmů jsou rovinné a prostorové transformace. První rovinná animace byla vytvořena již ke konci 60. let. Tato animace byla limitována prostředky, které v tehdejší době byly k dispozici. Nyní je animace řešena způsobem, že je modelován pohyb objektů v prostoru a ten je potom promítán do roviny. Pohyb jednotlivých objektů, které jsou objektem animace, je dán uzlovými body dráhy, která je posléze interpolována. Interpolace pohybu je následně doplněna skládáním rotačního a translačního pohybu. [10]

Jedním z požadavků při vytváření animace je pohyb po jiných než rovinných plochách. Nejčastěji to může být pohyb po kulové ploše. Čím více se při animaci snažíme dosáhnout přirozeného pohybu, tím méně nám vyhovuje prostá kinematika. V poslední době je v animaci řešen problém přirozeného vzhledu objektu. Rozmělnění hran, neostrých přechodů světla a stínu, neostré barvy atd. Při tvorbě animace je základem vytvoření snímku a jeho změna tak, aby se vytvořilo pomyslné zdání pohybu. Tento efekt je docílen především transformací rovinných obrázků nebo transformací ve 3D. [10]

1.7.3. Počítačová animace

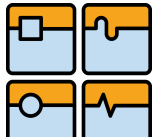
Počítačová animace je proces, který slouží pro tvorbu animovaných obrázků za využití počítačové grafiky. Při tvorbě počítačové animace se používají speciální programy. Existují dva druhy počítačové animace dvourozměrná a trojrozměrná. Většina programů, které slouží pro vytváření animací, vytváří animaci pomocí klíčových snímků. Jednotlivé pozice animovaného objektu je následně určena uživatelem pomocí těchto klíčových snímků a program sám dokáže pohyb dopočítat mezi uživatelem zvolenými pozicemi. Příklad tvorby počítačové animace za pomoci programu Motion Capture je znázorněn na obr.5.



Obr. 5. Příklad vytvoření animace v programu Motion Capture [11]

U dvourozměrné (plošné) animace se na začátku nejprve pomocí vhodně zvoleného programu vytvoří obrázek, který má určitý tvar a barvu. Obrázek, který je objektem animace, se poté umístí na obrazovku. Pro vytvoření dojmu, že daný objekt například padá, bychom potřebovali na obrazovce vykreslit tento objekt v každé fázi jeho pohybu, což zabere opravdu mnoho času. Ve vhodném programu stačí zadat pouze hlavní body, ve kterých se má tento objekt nacházet, případně můžeme zadat i trajektorii, po které se má pohybovat a snímky, které bychom potřebovali pro vytvoření animace, se automaticky mezi tyto body doplní. [12]

Při vytváření prostorové animace modelovací software vytvoří imaginární trojrozměrný prostor, ve kterém může návrhář pracovat. V trojrozměrné animaci je vytvářen matematický model trojrozměrného objektu tak, aby realisticky zobrazoval objekty i s jejich hloubkou. Vytváření 3D animace probíhá ve třech fázích a to modelování, animace a rendering. Zatímco dokončení modelování a renderingu trvá někdy opravdu déle, animace 3-D objektů je mnohem snadnější než u plošné animace obrázku. Zatímco dvojrozměrnými obrázky můžeme pohybovat jen dvěma hlavními směry nahoru a stranou (osa y a osa x), u třírozměrných modelů je přidána navíc ještě hloubka (osa z). Návrhář nejprve vytvoří drátěný model animovaného objektu. Nakreslí jednoduchý obrázek zadáním souřadnicových bodů nebo vektorů.

	Ústav výrobních strojů, systémů a robotiky	Str. 19
	DIPLOMOVÁ PRÁCE	

Jakmile definuje osu z, objeví se 3-D objekt mající určitou hloubku a objem. Drátěný model může vytvořit též zkombinováním základních tvarů (koule, válec, hranol apod.), které jsou v daném programu k dispozici.

Stavba původních trojrozměrných modelů je náročná na čas, ale dá se urychlit použitím již hotových drátěných modelů. Nebo může použít trojrozměrný digitalizér a udělat počítačový model ze skutečného objektu – návrhář se přitom dotýká různých bodů objektu perem, pozice pera je přesně změřena a přenesena do počítače. Po nasnímání dostatečného počtu bodů je možné postavit celý model v počítači. Složitý objekt může mít tisíce souřadnicových bodů, zvláště když simuluje reálný objekt, jako je například člověk. [12]

1.8. Počítačová grafika

Pod pojmem počítačová grafika se rozumí z technického hlediska obor informatiky, který využívá počítače pro vytváření umělých grafických objektů a dále také na úpravu zobrazitelných a prostorových informací, nasnímaných z reálného světa (například digitální fotografie a jejich úprava).

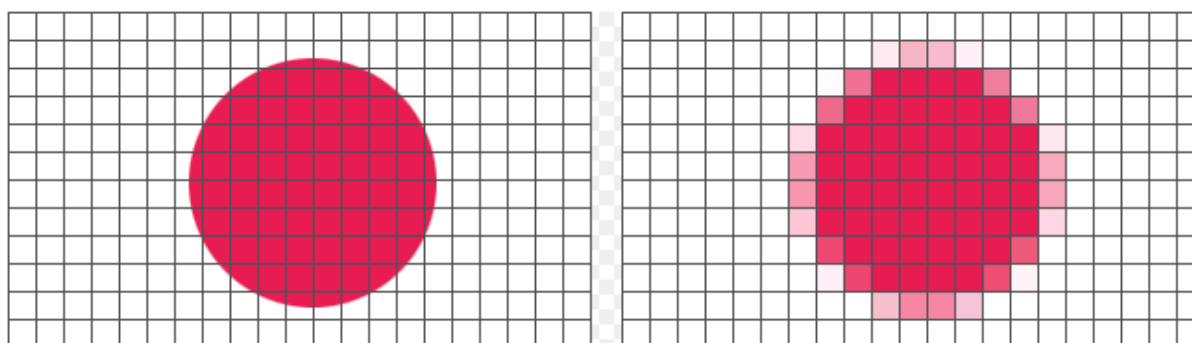
Z hlediska umění se jedná o samostatnou kategorii grafiky. Za autora spojení „počítačová grafika“ je považován William Fetter, designér firmy Boeing, který tento termín poprvé použil v roce 1960 při popisu své práce. Na počátku grafických technologií byly takové projekty jako Whirlwind, což byl první počítač, který pro výstup dat využíval CRT obrazovku a navíc umožňoval využití světelného pera jako vstupní jednotky.

Významným krokem byl počítač TX-2 vyvinutý roku 1959 v Lincolnově laboratoři v Massachusetts Institute of Technology. V roce 1963 pro tento počítač Ivan Sutherland naprogramoval první program využívající grafické možnosti počítače a tedy i první program s grafickým uživatelským rozhraním s názvem Sketchpad. Brzy se také významné počítačové firmy začaly zajímat o grafiku a v roce 1965 uvedla firma IBM na trh grafický terminál IBM 2250, první finančně dostupný grafický počítač. Koncem šedesátých let se konaly první konference a vznikly první obecně přijímané standardy, především díky organizaci SIGGRAPH. Od roku 1973 se pořádají pravidelné výroční konference této organizace, které se staly veletrhem novinek v oblasti počítačové grafiky. Ke konci sedmdesátých let se začaly rozšiřovat možnosti osobních počítačů a s nimi i způsoby praktického využití počítačové grafiky. Od osmdesátých let se v počítačových systémech využívají symboly, ikony, obrázky a další grafické prvky označované jako grafické rozhraní, které slouží ke snadnější a příjemnější komunikaci mezi uživatelem a počítačem. Konce tohoto desetiletí se také 3D grafika stala skutečností na SGI počítačích, který byly později využívány při tvorbě krátkých filmů ve studiu Pixar. Devadesátá léta přinesla nárůst popularity 3D grafiky především díky počítačovým hrám a animovaným filmům. V této diplomové práci budeme využívat klasickou 2D počítačovou grafiku, kterou lze rozdělit na dvě základní kategorie a to rastrovou a vektorovou grafiku, která je podrobněji popsána v následující části. [6]

1.8.1. Rastrová grafika

Rastrová jinak také známá pod názvem bitmapová grafika pracuje s obrazy jako s mřížkou jednotlivých barevných bodů neboli pixelů. Tyto body nejsou umístěny náhodně, ale jsou uspořádány do jisté mřížky, které se říká rastr. Díky tomu vzniklo pojmenování rastrová grafika.

V daném rastru mají body přiřazeny přesnou polohu i barvu např. RGB. Celková kvalita obrazu je pak závislá především na rozlišení a barevné hloubce. S rostoucím počtem bodů obrazu roste i zároveň jeho kvalita. V rastrovém obrázku se využívají body s různými odstíny šedé barvy, které zjemňují okraje a vytváří iluzi jako by se jednalo o jednotlivou barvu.



Obr. 6 Převod obrazu do bitmapové grafiky [5]

Bitmapová grafika je vytvářena rastrováním. Předloha je snímána za pomoci skeneru, digitálního fotoaparátu či kamery. Tento obraz v nekomprimované podobě obsahuje takové množství bodů, které se dá získat vynásobením počtu bodu na šířku a výšku.

Tyto obrázky se následně dají upravovat. Kvůli značné náročnosti bitmapových obrázků na obsazený prostor je často použita komprimace a kódování. Komprimace ztrátovým způsobem funguje způsobem, že jsou z obrázku odstraněny informace, které lidské oko není schopné postřehnout. Ostatní informace jsou kódovány. Komprimace bezztrátová využívá pouze kódování. Nejčastější formáty bitmapové grafiky jsou vypsány v tab. 1.

Nekomprimované	Komprimované	
	Bezeztrátové	Ztrátové
PCX	PCX	PCX
TIFF	TIFF	JPEG
BMP	GIF	JPEG 2000
	HDP	HDP
	MNG	
	PNG	
	APNG	

Tab. 1 Formáty bitmapové grafiky [5]

1.8.2. Vektorová grafika

Je to méně známý typ počítačové grafiky, jelikož programy pracující s tímto druhem nejsou ve velkém měřítku rozšířeny mezi běžnými uživateli. Na spoustě dnešních webech se využívají vektorové obrázky, které mají oproti bitmapovým tu výhodu, že je lze zkomprimovat na menší velikost a lze měnit jejich velikost beze ztráty kvality.

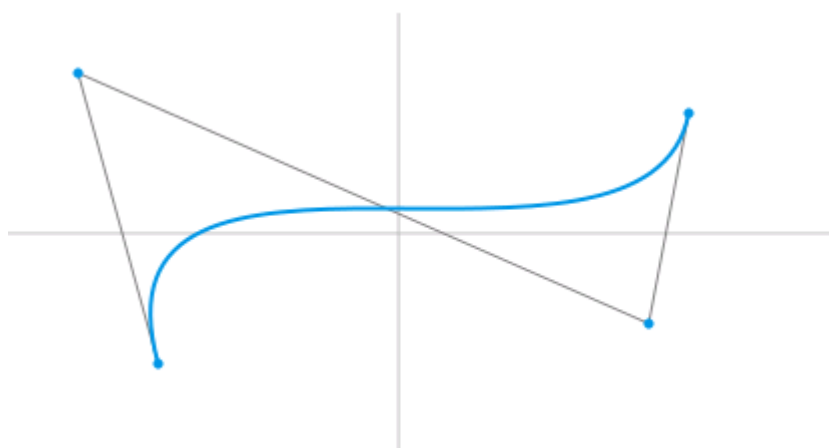
Tento typ obrázků je založen na matematických výpočtech, které umožňují vyšší stupeň komprese. Obraz je reprezentován za pomoci geometrických objektů (křivky, kružnice, lomené čáry, mnohoúhelníky).



Obr. 7 Ukázka efektivity vektorové grafiky při zvětšování: (a) originální obrázek; (b) zvětšeno jako vektorový obrázek; (c) zvětšeno jako rastrový obrázek [5]

Při změně velikosti se provádí výpočet a tím je vyloučeno zhoršení kvality. Každý bod na obraze je vyjádřen matematickým způsobem to znamená, že křivka takto nakreslená je definována souřadnicemi počátečního bodu, vektorem (odtud vznikl název grafiky), který určuje směr a zakřivení, a koncovým bodem. K rovnici se ještě přidají informace o barvě čáry a tloušťce. Vektorový obraz je zpravidla tvořen z mnoha menších a jednodušších objektů. Při vytváření složitějších tvarů se provádí buď spojováním a rozdělováním jednoduchých tvarů do výsledného složeného objektu nebo přímo kreslením a editací křivek. Složitější křivky jsou vytvořeny více jednoduchými v koncových bodech pospojovanými úseky.

Vektorový objekt je získatelný například kreslicími nástroji a je bez ztráty kvality po celou dobu práce s objektem. Takto získaný objekt je tvořen obrysem, ohraničením a výplní. Změna tvaru takového objektu lze provádět pouze změnou obrysu, výplň se do objektu generuje automaticky. Z kreslicích bych rád upozornit například na Béziovu křivku, která je popsána za pomoci čtyř bodů – dvou krajních, tzv. kotevních, které určují začátek a konec úsečky, a dvou kontrolních určujících vlastní tvar této úsečky. Francouzský matematik Pierre Bézier vyvinul metodu, díky které je schopen popsat pomocí čtyř bodů libovolný úsek křivky.



Obr. 8 Bézierova křivka. [5]

Vektorová grafika se používá nejčastěji v počítačové sazbě při tvorbě ilustrací, diagramů, animací. Při práci s vektorovou grafikou se využívají speciální vektorové editory (např. Adobe Illustrator, Corel Draw, Inkscape, Zoner Callisto). Nejčastější formáty vektorové grafiky jsou popsány v tab. 2.

Přípona formátu	Vektorový editor
EPS, PS	PostScript
PDF	Portable Document Format
AI	Adobe Illustrator Artwork
CDR	Corel Draw
SVG	Scalable Vector Graphics
ZMF	Zoner Callisto

Tab. 2 Formáty vektorové grafiky. [5]

1.9. Software a technologie využité při vývoji

V této podkapitole diplomové práce je blíže popsán software a technologie, které byly využity při zhotovení praktické části. Nachází se zde také stručné shrnutí některých základní technologické prostředky, které při zhotovení byly použity.

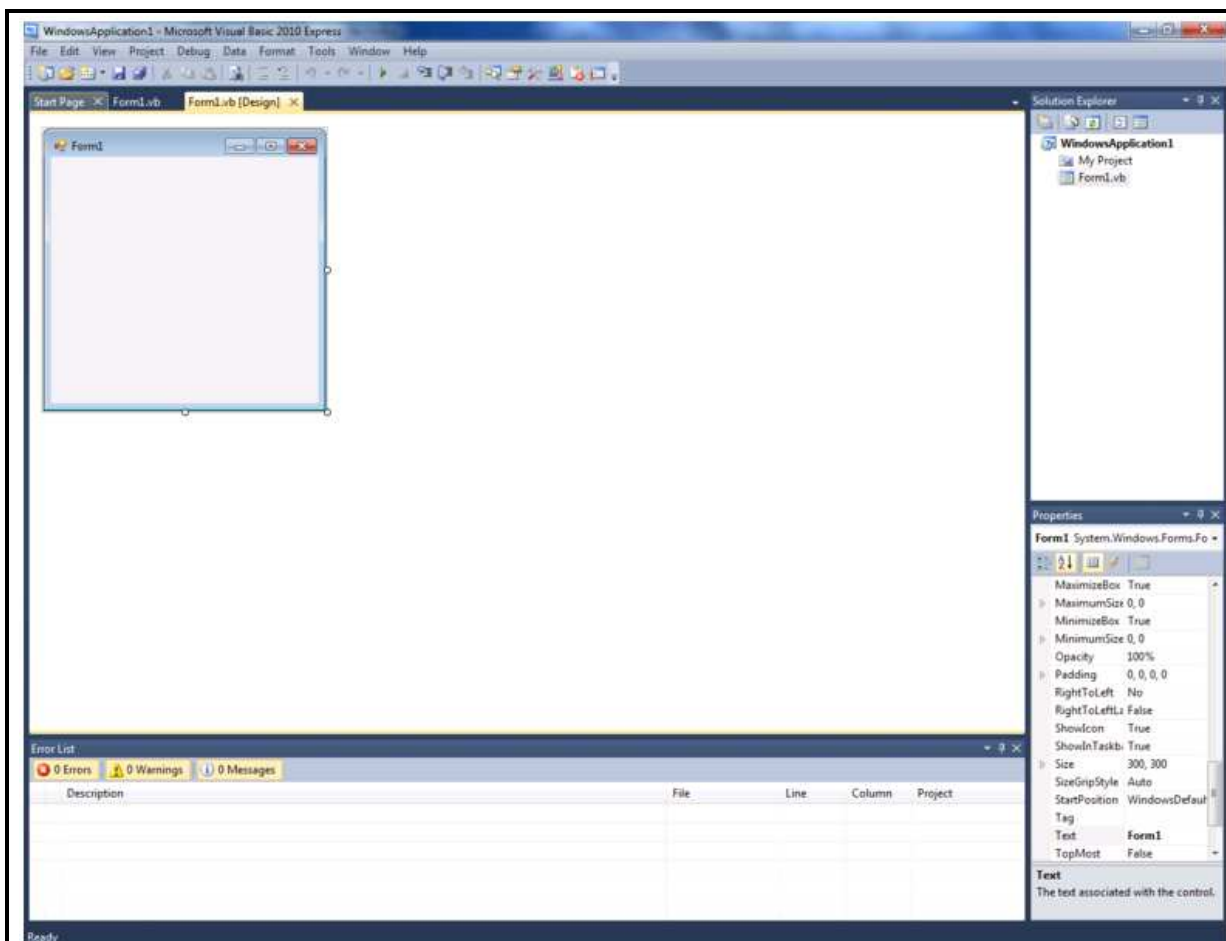
1.9.1. *Microsoft Visual Studio .NET*

Pro vytvoření praktické části diplomové práce bylo zvoleno programovací prostředí Microsoft Visual Studio.Net. Jedná se víceúčelové programovací prostředí, které bylo vyvinuto, jak již název vypovídá společností Microsoft. Toto programovací prostředí je vybaveno mnoha jazyky např. C++, Visual Basic, C# a implementace Microsoft SQL, které můžeme použít pro vytváření výsledných aplikací.

Kromě klasických placených verzí Standart či Professional lze od roku 2005 sehnat i volně dostupnou verzi Express Edition, která je oproti placeným verzím ochuzena o některé funkce, ovšem pro účely diplomové práce je plně použitelná. V současné době je možné získat volně i verzi Professional, ovšem pouze jako 90 denní demoverzi, což jsem nepovažoval pro zhotovení praktické části práce za reálné (proto bude pozornost v textu nadále omezovat pouze na Express Edition).

Jelikož diplomová práce bude vypracována v jazyce Visual Basic .NET omezíme se také z celého programovacího prostředí pouze na části zabývající se tímto programovacím jazykem.

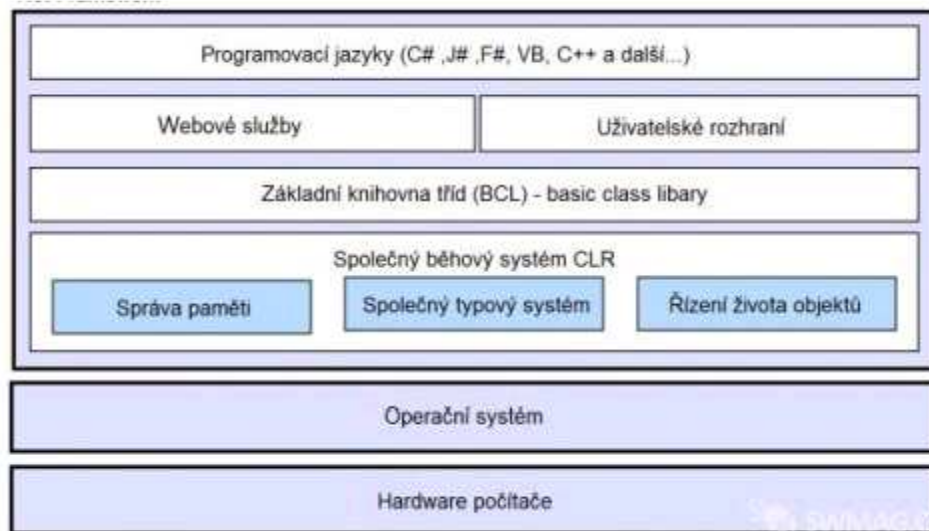
Express Edition je oproti ostatním placeným verzím rozdělena do modulů. Každý z těchto modulů si instaluje pouze ty doplňky, které sám nutně potřebuje pro svoji funkci. Pro funkci našeho modulu je zapotřebí například Microsoft Visual Studio .NET Framework , který je základním stavebním kamenem pro tvorbu aplikací na platformě Windows a je tedy i základem pro vytvoření praktické části této diplomové práce. Pro bližší představuje je znázorněno návrhové prostředí na obr.9. Na obrázku je znázorněn čistě projekt Windows Form Aplicaton v programu Visual Studio .NET.



Obr. 9 Pracovní prostředí Microsoft Visual Studio 2010 Express

1.9.2. .NET FRAMEWORK

.NET Framework je rozsáhlá softwarová platforma, která je určena pro vývoj různých druhů aplikací. Struktura této platformy je tvořena třemi základními vrstvami. První, známější vrstva se skládá z tzv. Common Language Runtime a Basic Class Library. Do první části spadají programovací jazyky jako například C#, J#, F#, VB, C++ apod. Tato část slouží k běhu systému a skládá se z webových služeb a uživatelského rozhraní. V druhé části je možné nalézt společný běhový systém CLR, ve kterém se dále nachází správa paměti, společný typový systém a řízení života objektů. Dalším ze základních vrstev je operační systém a v neposlední řadě se zde také nachází hardware počítače. Schéma struktury .NET Framework je pro názornost zobrazeno na obr.10 . [8]



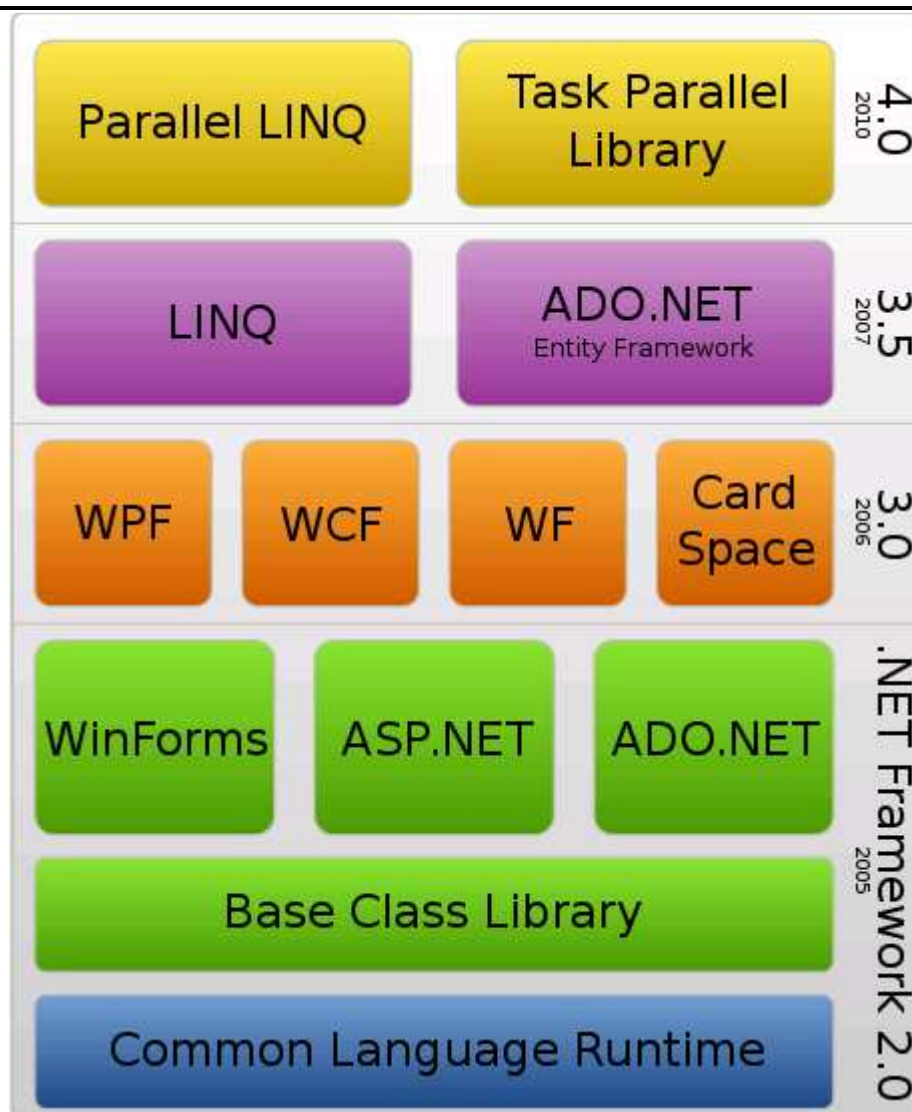
Obr. 10 Schéma struktury .NET Framework [8]

Common Language Runtime

Tuto vrstvu si můžeme představit jako nějaký virtuální stroj, ve kterém pracují běhové funkce této platformy a programovací jazyky zde mají k dispozici všechny knihovny tříd systému, se kterými jednodušeji pracují. Tento systém obsahuje obdobu strojového kódu, který se nazývá Microsoft Intermediate Language. Kód není kompilován přímo do strojového kódu, ale do MSIL kódu, což znamená omezení výkonnosti prostředí. Tímto nedostatkem jsou vynahrazovány mnohé výhody, jako například řízený kód, jednotný typový systém či správa paměti. Mezi největší výhody této architektury patří schopnost vývoje aplikací v mnoha různých programovacích jazycích. Tato vlastnost se nazývá tzv. Cross-language interoperability, což znamená možnost spolupráce mezi jednotlivými jazyky. [8]

Microsoft Intermediate Language

Jazyk MSIL, který má mnoho společného s assemblerem, byl vytvořen tvůrci programovacích jazyků za účelem vyvíjení nových jazyků pro platformu .NET. Při kompilaci programu v systému .NET vezme překladač na vstup kód příslušného jazyka a výstupem bude práce aplikace v MSIL. Poté nám CLR zajistí, že při spuštění aplikace se program v MSIL sám přeloží do základního strojového kódu. [8].



Obr.11 Vývoj .NET Frameworku v průběhu let [6]

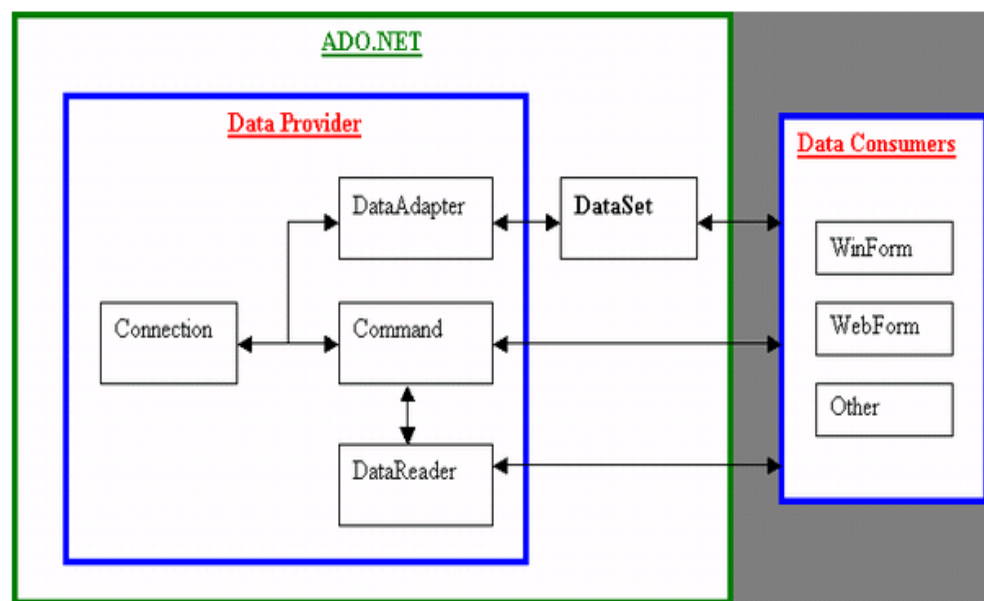
1.9.3. ADO .NET

Praktická část diplomové práce je založena na práci s daty, které jsou uloženy v databázi Microsoft Access. Proto bych zde stručně popsal technologii, která byla použita pro přístup a práci s těmito daty. ADO.NET (Microsoft ActiveX Data Objects .NET) je rozsáhlou knihovnou tříd, které nám umožňují pracovat s databázemi a datovými soubory. Jelikož se jedná o součást .NET Frameworku, poskytuje ADO.NET uživateli veškeré vymoženosti, které obsahuje samotný .NET Framework. Jde zejména o podporu mnoha jazyků (C# , Managed C++, Visual Basic .NET, J# a dalších) automatické uvolňování paměti a objektově orientovaný přístup. ADO.NET je jedna z množiny tříd nabízejících služby pro přístup k datům a tvorbu databázových aplikací. Daty mohou být chápány převážně informace uložené

v databázi. Ať již se jedná o data v databázích například na Microsoft SQL Serveru nebo data zpřístupněná přes OLE DB nebo XML.

Mezi jeho hlavní přednosti patří především jednoduchý způsob použití a rychlost při zpracování. Stačí vytvořit spojení se serverem, se kterým budeme chtít pracovat, pomocí zvoleného adaptéru a zadaného dotazu získat z databáze data a ty pak načíst do některé z připravených konstrukcí, které slouží pro práci s daty z tabulek. Nástroje ADO.NET byly navrženy tak, aby se oddělil způsob přístupu k datům od manipulace s daty. K první skupině patří .NET Framework data provider obsahující množinu komponent zahrnujících podmnožiny *Connection* (připojení), *Command* (množinu příkazů pro vybrání dat), *DataReader* (načítání dat) a *DataAdapter* (adaptér pro připojení k databázi). Ve druhé skupině je možné nalézt objekty jako *DataSet* (skládající se z objektů *DataTable*, *DataRow*). Jsou to objekty, které uchovávají data načtená z databází. Tyto objekty mohou s daty pracovat podobným způsobem jako přímo s daty v databázi.

Na první pohled by se mohlo zdát, že se jedná pouze o upravenou existující verzi ADO (ActiveX Data Objects) pro platformu .NET, avšak tyto dvě struktury nemají mnoho společného. Hlavní odlišností je, že ADO.NET patří mezi nativní databázová rozhraní zatímco ADO můžeme řadit mezi rozhraní obecná. ADO.NET podobně jako ADO umí komunikovat s jakýmkoliv zdroji dat, tzn. relačními i nerelačními. ADO.NET je obsažen v .NET Framework a skládá se, podobně jako jiné části .NET Framework, z množiny tříd, které mohou být ve vzájemné interakci. Tyto třídy bychom mohli rozdělit na Data Provider (poskytovatel dat), který má jako hlavní úkol komunikaci s fyzickým uložištěm dat a DataSet (sadu dat), která obsahuje skutečná data. Obě tyto části komunikují s Data Consumers (konzumenti dat), kterými mohou být WinForm nebo WebForm.

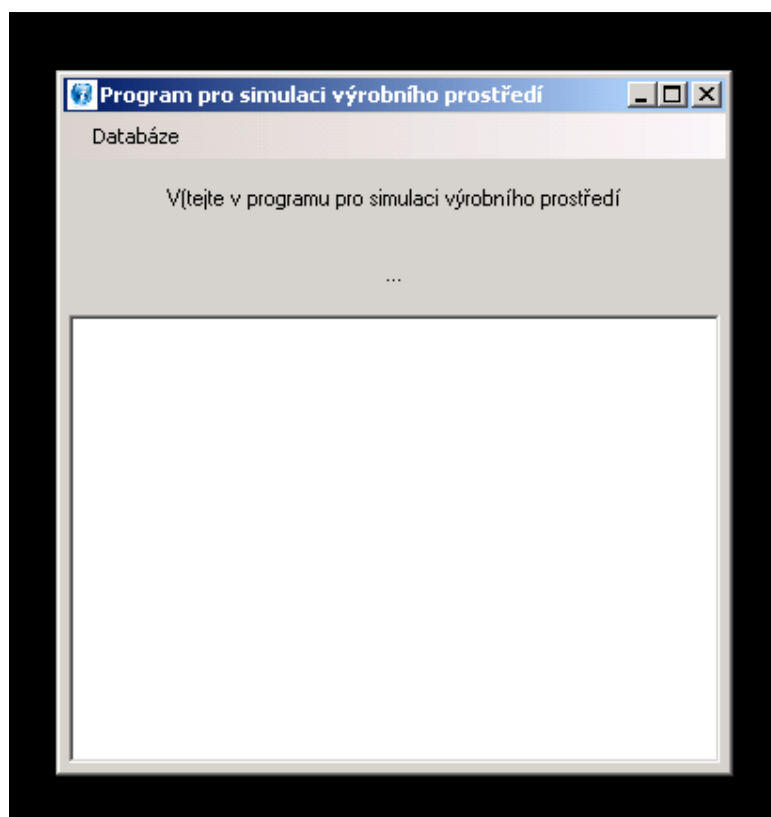


Obr.12 Objektový model ADO .NET [9]

2. Praktická část

V praktické části diplomové práce bych rád přiblížil postup při vytváření programu pro animaci výrobního procesu, popsal jeho základní podobu a funkce, které jsou v tomto programu obsaženy. Na výběr byla možnost zhotovení programu ve 2D nebo ve 3D variantě. Zvolil jsem 2D realizaci daného problému, jelikož mi tato varianta byla bližší. Základem pro vytvoření praktické části je, jak již bylo řečeno, práce s daty, které jsou obsaženy v databázi Microsoft Access. Jako vstup do animačního programu mi byl poskytnut vedoucím diplomové práce zkušební soubor pod názvem anim.mdb, který je uložen spolu s programem na CD. V této databázi jsou obsaženy data o proběhlé animaci výrobního procesu a úkolem praktické části je tyto data předpřipravit a následně zpracovat.

Při vytváření daného programu bylo základní snahou vytvořit univerzální nástroj, který by byl schopen takto animovat, jakoukoliv další nově vytvořenou databázi. Samozřejmostí však je podobná struktura databáze, kterou bude program zpracovávat. Mohou se měnit pouze data, která jsou uvnitř této databáze. Pod pojmem data v databázi můžeme chápat množství a druhy jednotlivých použitých strojů, jednotlivé časy proběhlé simulace, měnící se zakázky atd. Výstupem praktické části bude tedy externí program pro grafické zpracování v minulosti proběhlých simulací. Úvodní okno animačního programu je znázorněno na obr.13.

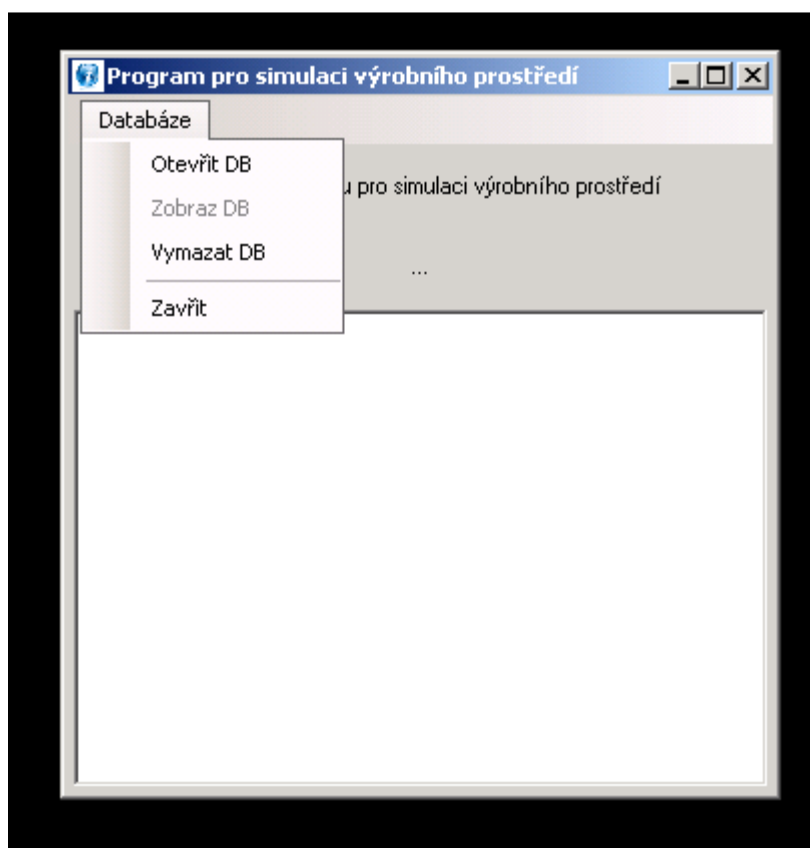


Obr.13 Úvodní okno animačního programu

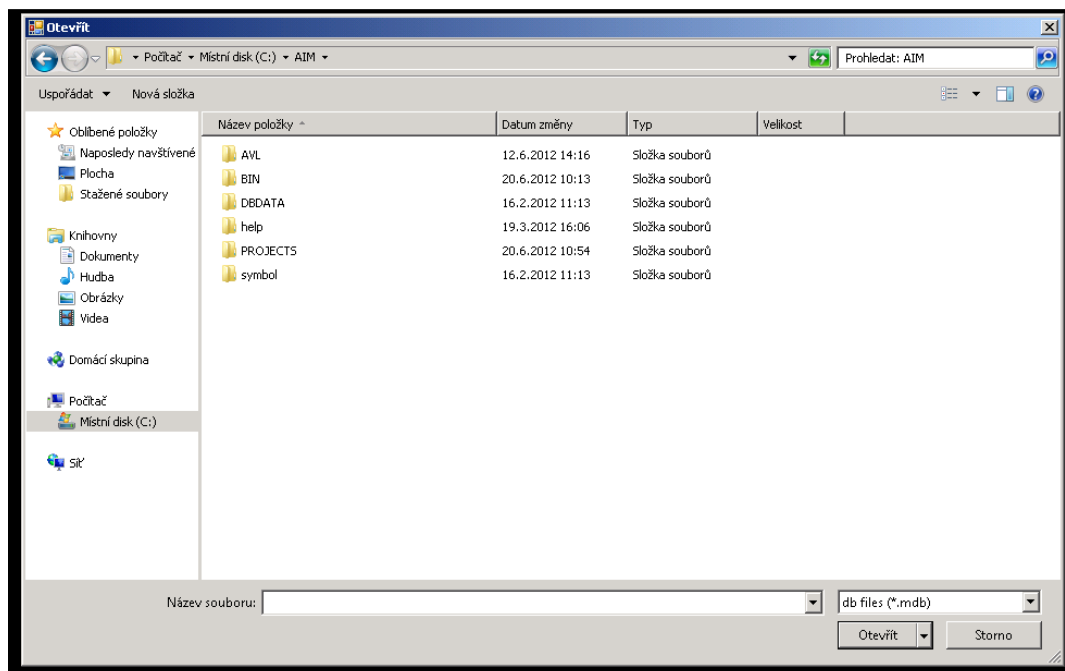
2.1. Popis a realizace vybavení pro animaci výrobních systémů

Úvodní okno programu je vybaveno základními komponentami, které nám visual basic poskytuje. Jedná se o komponenty menustrip, které slouží v programu jako hlavní panel, který je běžně používán ve většině aplikací. Následně nástroj label, ve kterém je napsaná první přivítací hláška programu a v neposlední řadě nástroj listbox, který slouží pro vypsání jednotlivých zakázek zkušebního souboru. Po vybrání požadované databáze jsou jednotlivé zakázky zobrazeny právě v tomto listboxu a můžeme následně přistoupit touto cestou ke spuštění pracovního prostředí, kde probíhá animace výrobního procesu. Celá úvodní obrazovka podobně jako i zbytek programu je zarovnán do tzv. docku, který zajišťuje zvětšování jednotlivých komponent při manipulaci s velikostí nadřazeného okna. Tímto nástrojem tak lze snadno zabránit tomu, aby se aplikace rozjížděla do nepatřičných tvarů při manipulaci s velikostí.

Po stisknutí položky Databáze na hlavním panelu se zobrazí seznam možnosti, které má uživatel k dispozici při práci s programem. Tento seznam je znázorněn na obr 14. Pro klasického uživatele této aplikace bude nejspíš hlavní položka Otevřít DB, která po stisknutí vyvolává nástroj OpenFileDialog. OpenFileDialog je znám zřejmě každému klasickému uživateli windows a pro názornost je zobrazen na obr.15.



Obr.14 Seznam základních možností programu



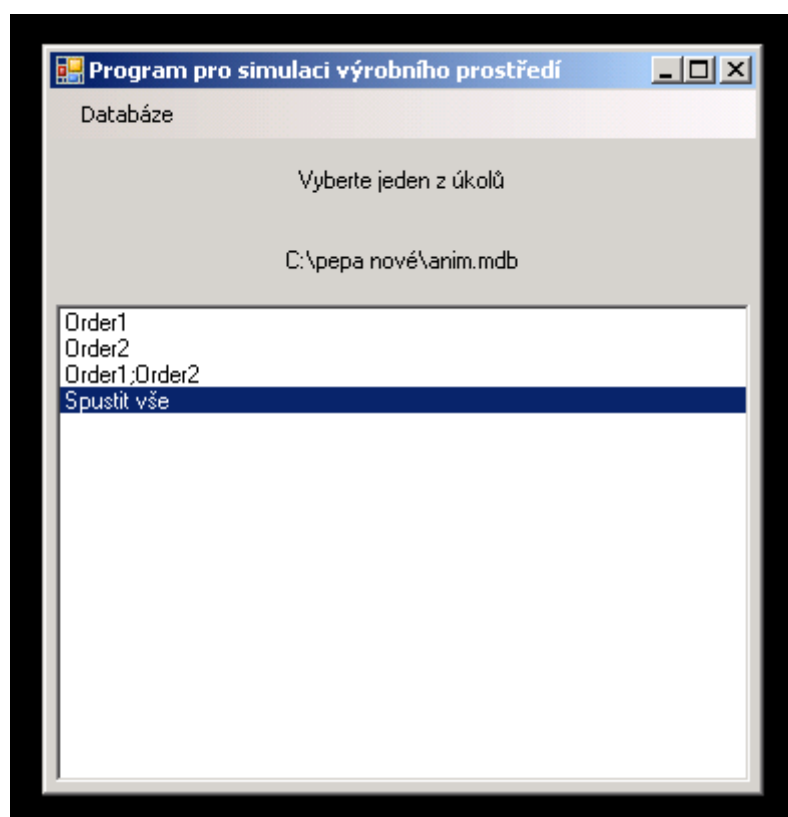
Obr. 15 Vyvolání OpenFileDialog pro vyhledávání souboru Microsoft Access

Tento OpenFileDialog poskytuje uživateli nástroj pro vyhledání a výběr vstupního souboru. Z obrázku si je možné všimnout, že filtrování jednotlivých souborů v uživatelském počítači je nastaveno pouze na soubory databáze Microsoft Access pro snadnější a rychlejší vyhledávání. Po vybrání požadované souboru se objeví uživateli na obrazovce tzv. msgbox, který slouží pro informaci o načtení databáze s její aktuální polohou na disku. V tomto kroku probíhá i požadovaná synchronizace (předpříprava) zkušebních dat pro následné zpracování programem. Prvním krokem této synchronizace je nahlédnutí do námi otevřené databáze a kontrola existence tabulek (RESRC000, PART, ABOPTS, POOL000 atd.).

Pokud tyto tabulky neexistují nebo jsou nějakým způsobem poškozené, pak se vytvoří zcela nové tabulky podle předem připravené struktury, která je obsažena v programu. V další fázi synchronizace probíhá kontrola obsahu těchto tabulek s původními tabulkami zkušební souboru. Kontroluje se obsah tabulky Ressource vůči tabulce RESRC000. Pro lepší pochopení bych rád uvedl, čím tyto tabulky mohou být naplněny. Tabulka Ressource obsahuje jednotlivé informace o strojích, které byly využity v animaci výrobního procesu. Můžeme zde najít jednotlivé názvy strojů a například i informace o tom, jestli tyto stroje pracují ve skupině nebo samostatně. Uvádím tento detail kvůli tomu, protože v naší zkušební databázi se právě stroje pracující ve skupině vyskytují. Tyto údaje o strojích jsou přeneseny do tabulky RESRC000 a při každém spuštění probíhá kontrola, jestli tabulka RESRC000 a Ressource obsahuje ty samé informace o strojích, přičemž tabulka

Ressource je řídící. Následně je pro jednotlivé stroje vytvořen v tabulce POOL000 sklad, který stroje mohou používat pro odkládání zpracovaných dávek.

Nastane-li situace, při které se objeví stroj pouze v tabulce RESRC000 a není obsažen v tabulce Ressource, pak se tento stroj z tabulky RESRC000 vymaže a s ním i údaje, které by se mohly vyskytnout v tabulce POOL000. Obdobná synchronizace probíhá i u ostatních tabulek, které nejsou součástí původní zkušební databáze a jsou do ní doplněny programově. Po dokončení této úvodní přípravy dat zkušebního souboru se naplní listbox hlavního okna programu možnostmi, na kterých můžeme nyní již demonstrovat animaci.

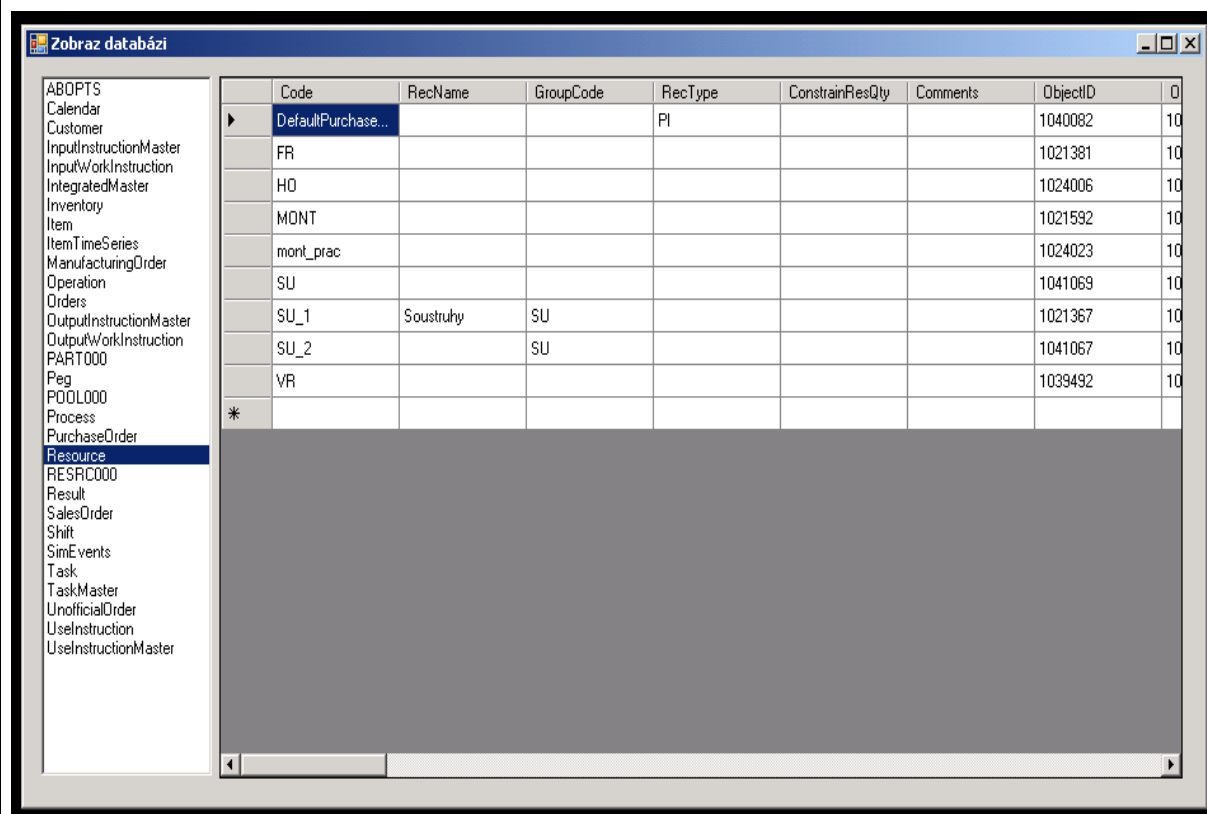


Obr. 16 Úvodní okno aplikace po načtení zkušebního souboru

Zpracovaná data pro následnou demonstraci animace výrobních systémů jsou názorně uvedena na obr. 16. V tomto kroku je možné zapnout animaci pro průběh jednotlivých dávek výrobním systémem. Jedná se o zakázky Order1 a Order2 a nebo přímo tyto dvě položky animovat naráz. Samozřejmostí je možnost animace celého výrobního procesu, který se zobrazí při stisknutí položky Spustit vše. Bylo řečeno, že funkce Otevřít DB v menu hlavního panelu, slouží spíše pro klasického uživatele této aplikace. Naproti tomu možnost zobrazit databazi slouží spíše pro uživatele, který má již určitou představu o tom, jak má struktura vstupní databáze do programu vypadat. Tato funkce totiž umožňuje rychlou kontrolu obsahu zkušebního souboru. Po spuštění funkce zobrazit DB se vyvolá nový formulář, který obsahuje opět základní

komponenty. Listbox a DataGridView jsou zde nyní použity v roli prohlížeče Microsoft access databází. V listboxu jsou zobrazeny jednotlivé tabulky zkušební souboru a datagridview zde zobrazuje jednotlivé sloupce vybraných tabulek. Je samozřejmostí, že před použitím této funkce programu, musí být načtenen zkušební soubor. Jinak se na hlavní obrazovce objeví pouze msgbox, který nás informuje o tom, že požadovaná databáze se nepodařila načíst. Náhled do zkušební souboru je zachycen na obr.17.

Poslední možností, která se nabízí v úvodní části programu je vymazání databáze, které vrátí zkušební soubor do původního stavu. Jsou smazány všechny nastavení, která si uživatel mohl během práce s programem vytvořit. Může se jednat například o nastavení obrázků pro jednotlivé stroje, polohy jednotlivých pracovišť atd. Samozřejmostí je, že hlavní nabídka obsahuje i položku pro ukončení celého projektu, na kterou je klasický uživatel windows zvyklý z každého programu.



Obr.17 Náhled do zkušební souboru pro animaci výrobních procesů

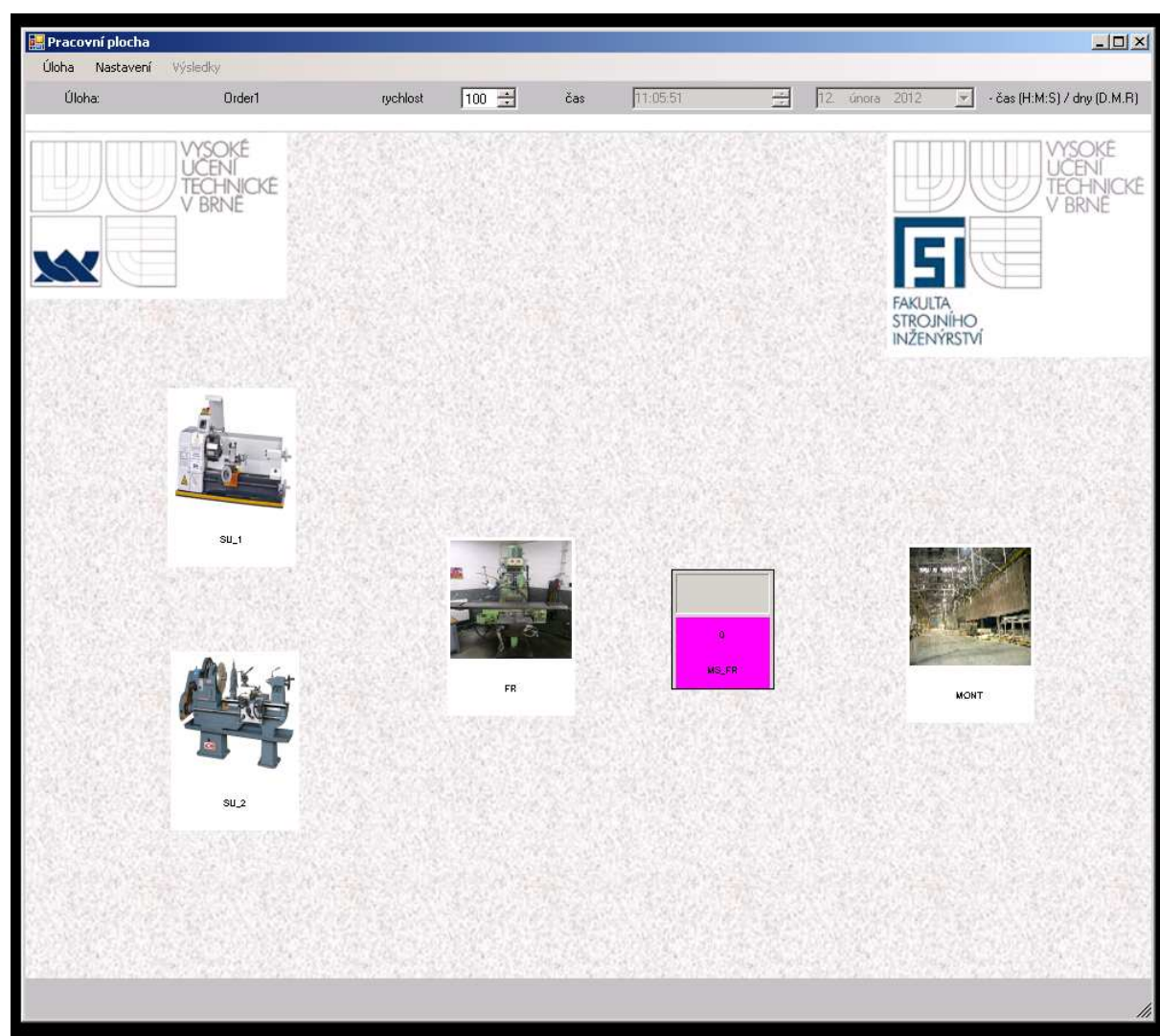
2.2.1. Realizace pracovního prostředí

Při návrhu pracovního prostředí diplomové práce, bylo bráno v potaz to, aby si uživatel sám mohl vybrat, jakou část výrobní zakázky chce animovat. Při kliknutí na

libovolnou z možností, které jsou uživateli nabídnuty v hlavním menu, se otevře opět zcela nový formulář s názvem pracovní plocha. V tomto formuláři stráví uživatel většinu času při práci s danou aplikací. Formulář obsahuje mnoho nastavení a možností, kterými je možno animaci různě graficky upravovat. Okno pracovního formuláře před spuštěním animace výrobní zakázky Order1 je zobrazeno na obr.18.

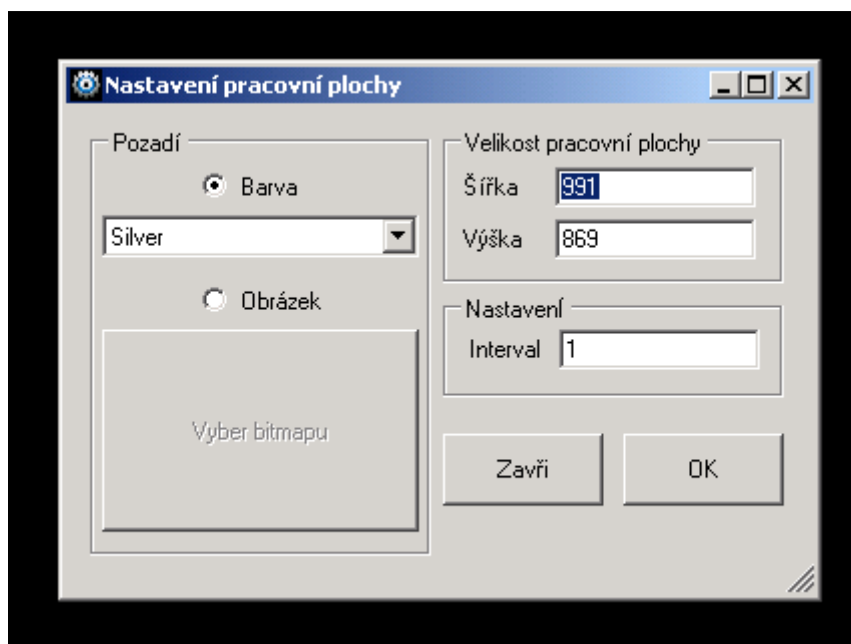
Oproti základnímu zobrazení, které se objeví uživateli při prvním načtení zkušebního souboru, je toto okno již znatelně přenastaveno. Při prvním spuštění animace libovolné zakázky jsou jednotlivé stroje a sklady rozmístěny v levém horním rohu na jednom místě a uživatel si je musí podle svého uvážení rozmístit tak, jak uzná za vhodné nebo jak zrovna potřebuje. Pro informaci stroje a sklady jsou zde tvořeny za pomoci komponent control, které poskytuje visual studio. Na pracovní ploše jsou nyní již také přednastavené obrázky pro jednotlivé stroje, které vytvářejí reálnější vzhled celé aplikace.

Pracovní prostředí pracuje pouze s objekty (stroje, sklady, které jsou používány v dané části výrobní zakázky.



Obr.18 Okno pracovního formuláře pro zakázku Order1

Jedním ze základních cílů této aplikace byla možnost vytvoření vlastního pozadí pro formulář pracovní plocha. Při prvním spuštění pracovní plochy nejsou některé možnosti zpřístupněny do té doby, než poprvé animaci spustíme. V hlavním panelu pracovní plochy se nachází položka nastavení, ve které můžeme právě nastavení pracovní plochy nalézt. Tato možnost je uživateli k dispozici hned po spuštění pracovní plochy. Po použití této funkce se spustí nový formulář s názvem nastavení pracovní plochy, který je pro představu zachycen na obr.19 .



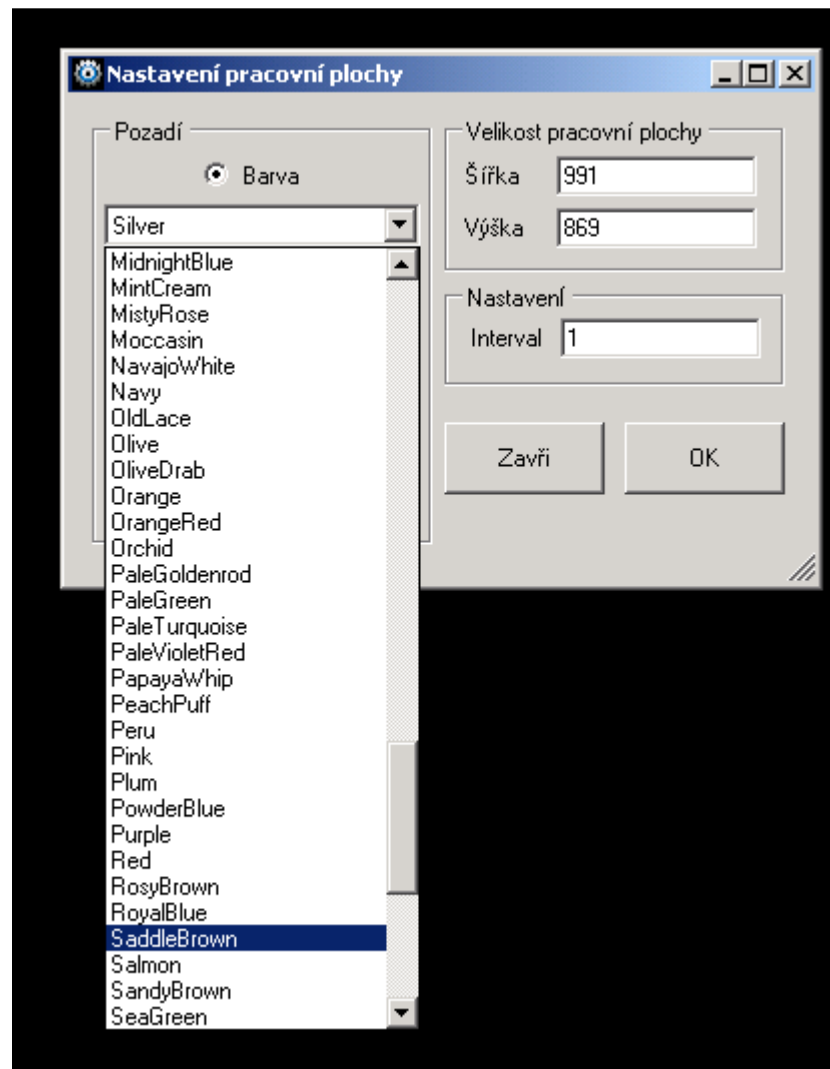
Obr.19 Formulář, který umožňuje nastavení pracovní plochy

Tento formulář slouží pro libovolné nastavení pracovní plochy. Okno je zhotoveno za pomoci základních prvků, které visual studio poskytuje. Mezi doposud nově použité objekty, které zde byly použity, můžeme zmínit snad GroupBox, TextBox ,Button ,ComboBox nebo RadioButon. Všechno jsou to ale zajisté objekty, se kterými se klasický uživatel počítače, již zajisté nejednou setkal. V daném okně je možnost nastavení pozadí své animace čistě ve formě barvy nebo zde uživatel může zvolit jinou variantu, kterou je možnost svého vlastního obrázku.

Při zvolení barvy se uživateli zpřístupní možnost výběru z opravdu nepřehledného množství barev, které tato aplikace poskytuje. Názorná ukázka je zobrazena na obr.20. Pokud se však uživatel rozhodne, že by jako pozadí animace radši použil např. pudorys dílny, ve které zkušební data pro animaci získal, stačí přepnout za použití RadioButon možnost obrázek. Při výběru vlastního obrázku pro pracovní plochu opět použijeme nástroj OpenFileDialog, kterým lze snadno dohledat cestu k vybranému souboru. Obrázek může být tvořen různými formáty, které jsou nastaveny ve filtru OpenFileDialogu.

Kdyby však opravdu nastala situace, že požadovaný obrázek neodpovídá žádnému ze zobrazených formátů, určitě v dnešní době existuje mnoho programů, které daný obrázek do odpovídajícího formátu svedou převést. Dále je zde možnost nastavení výšky a šířky pracovního formuláře a nastavení intervalu, který bych rád

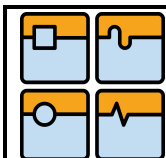
objasnil později. Při potvrzení tlačítkem se všechny nastavené hodnoty, které byly zvoleny, převedou do pracovního formuláře.



Obr.20 Vzorek barvy, kterou je možno nastavit jako pozadí pracovní plochy.

Před spuštěním animace je zde také možnost v hlavním panelu pracovního formuláře kliknout na položku Úloha, kde se nachází funkce například pro spuštění animace dané výrobní dávky. Restartování již dokončené animace pro možnost následného spuštění. Dále menu nabízí možnosti uložit, které všechny doposud nastavené hodnoty uloží do zkušební souboru, aby se uživatel k nastavenému rozhraní mohl v budoucnu kdykoliv vrátit. V poslední řadě je zde možnost zastavit animaci, která je nastavena, aby se zpřístupnila až při spuštění animace.

Dalším z cílených prvků programu byla možnost editace jednotlivých strojů a jejich skladů. Jak již bylo řečeno, při prvním spuštění programu jsou uživatelé všechny komponenty, se kterými bude v daném kroku pracovat, rozmístěny v levém horním rohu. Je dobré, aby měl uživatel představu o tom, jak bude výsledná animace zhruba vypadat a podle toho si rozmístil stroje a sklady na požadované pozice. Levé tlačítko myši slouží pro přesouvání strojů a skladu po pracovní ploše. Právě tlačítko



DIPLOMOVÁ PRÁCE

myši vyvolá editaci označených objektů. Při editaci stroje se zobrazí nový formulář nastavení stroje, který je zobrazen na obr.21 .

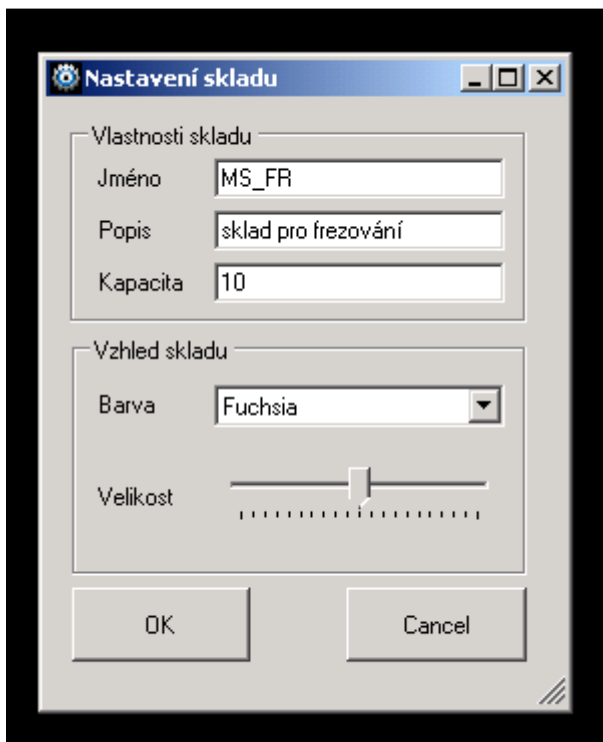
Obr.21 Okno, které slouží pro nastavení vlastností stroje.

V tomto okně je možnost změnit název vybraného stroje. Přidat k němu libovolný popis pro přesnější informaci o daném stroji. Je zde také možnost měnit velikost daného objektu, který stroj představuje. Pro nastavení velikosti stroje byl použit objekt TrackBar. Do hlubšího nastavení stroje se může uživatel dostat při stisknutí tlačítka nastavení ikonek.

Obr.22 Formulář, který slouží pro nastavení obrázků stroje

V tomto formuláři je k dispozici nastavení zvoleného obrázku pro vybraný stroj. Stroj může v průběhu animace své stavy měnit a proto je tedy možné nastavit obrázky pro všechny možné stavy, do kterých se vybraný stroj může v průběhu animace dostat. Nastavení vlastního obrázku pro stav stroje busy je znázorněn na obr.22. Za povšimnutí zde stojí malý černý čtvereček v levém horním rohu obrázku. Tento objekt není totiž součástí obrázku, ale slouží pro polohové znázornění zpracovávané dávky na stroji. Tato poloha lze tahem objektu měnit a určovat tak pro každý stroj zvlášť, kde se zpracovávaná dávka bude zobrazovat. Ve větších zakázkách se ve výrobním procesu vyskytuje množství různých dávek, které uživatel může od sebe oddělit barvou objektu reprezentující danou dávku. Po ukončení nastavení stroje požadovanými hodnotami je nutné opět provést uložení v hlavním panelu pracovní plochy, aby se údaje zaznamenaly do zkušebního souboru.

Formulář pro editaci skladů je o poznání jednodušší než formulář sloužící k editaci strojů. Nachází se zde opět možnost měnit jméno skladu nebo přidat popis pro upřesnění. Také je zde přidána možnost, která slouží pro zvětšování či zmenšování objektu, který vybraný sklad reprezentuje. Oproti editaci u strojů se zde nachází položka pro nastavení kapacity skladů a je zde také možnost změny barvy skladu. Okno, které slouží pro nastavení skladu je zobrazeno na obr.23 .

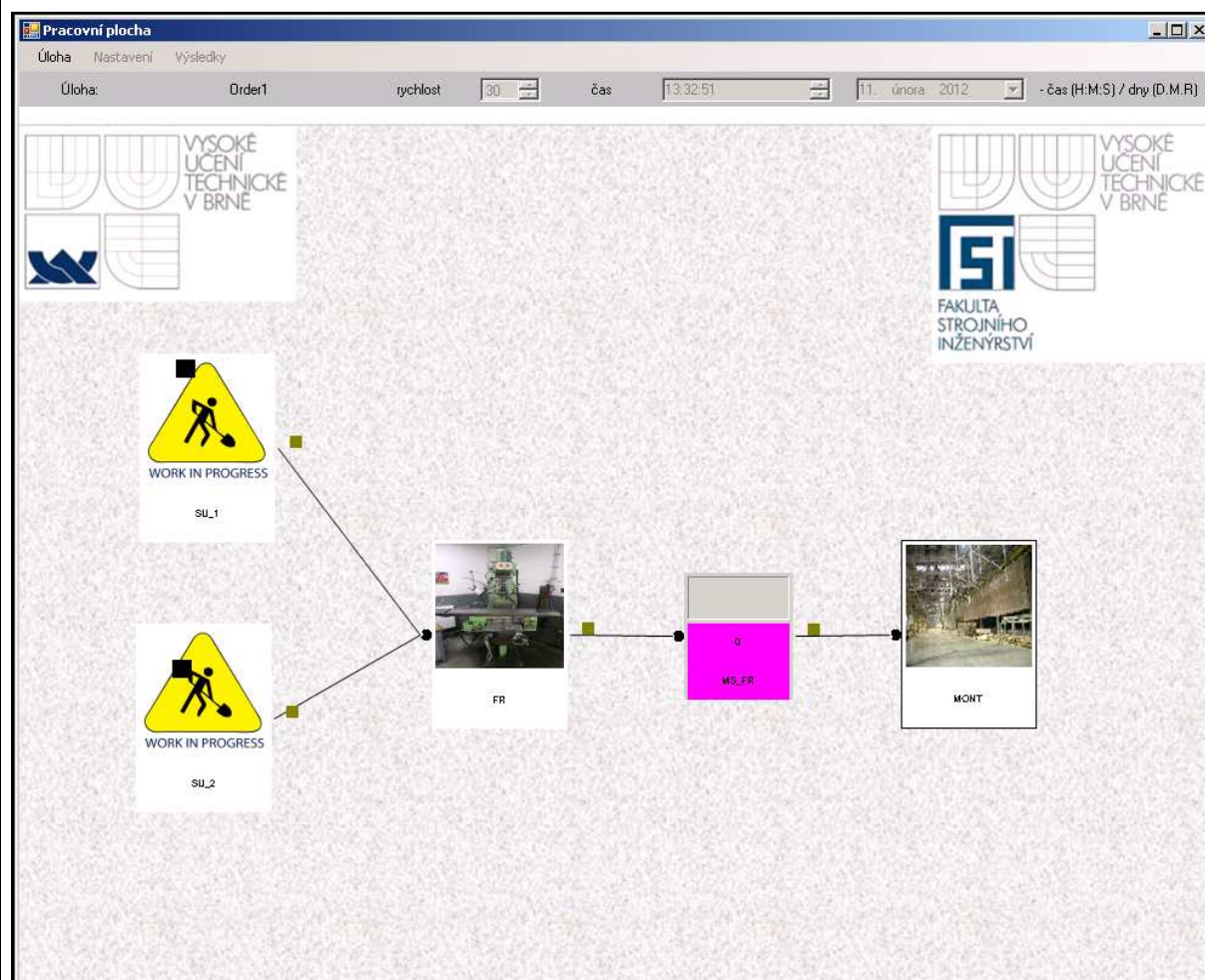


Obr.23 Formulář, který slouží pro nastavení skladu

Po dokončení všech předešlých nastavení byly vyčerpány možnosti, jak lze uživatelské rozhraní před animací graficky upravit. Zbývá už jen nastavit rychlost průběhu animace, která se nachází v hlavním okně pracovního formuláře a následně průběh výrobní zakázky spustit.

2.2.2. Popis průběhu animace výrobním systémem

Při prvním spuštění se na pracovní ploše vytvoří nový objekt linshape, který visual studio obsahuje. Tento objekt v animaci reprezentuje směr pohybu dávky výrobním systémem. Směr pohybu dávky měl být původně reprezentován šipkou, ale kvůli chybějícímu parametru šipky u objektu lineshape, jsem zvolil rovnou přímku zakončenou tečkou. Oproti klasickému programovému kreslení objekt lineshape poskytuje možnost přístupu přímo ke zvolenému prvku a změnu jeho parametrů za běhu aplikace, která u klasického vykreslování chybí. Okno pracovní plochy je pro názornost zobrazeno na obr.24.

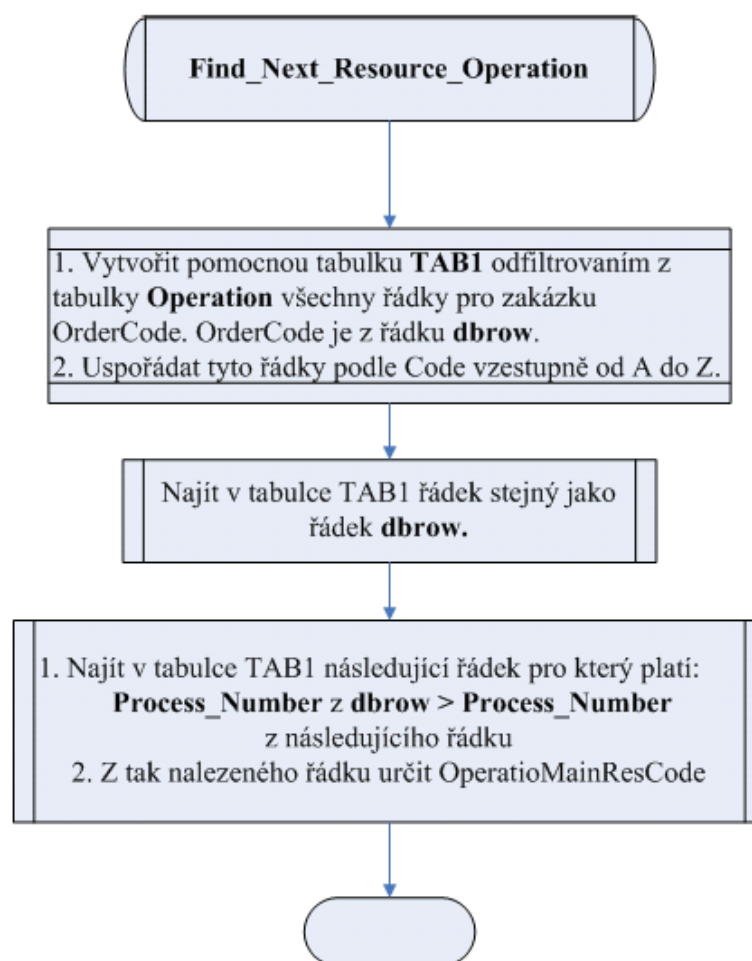


Obr.24 Okno pracovní plochy při spuštění animace

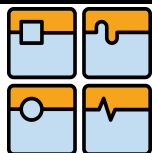
Objekt lineshape je řízen daty, které jsou obsaženy ve zkušebním souboru a podle toho jsou tímto prvkem propojeny jednotlivé stroje a sklady. V úvodní části zvolené zakázky se přepnou oba soustruhy do pracovní polohy. Na obr.24 je možno vidět odlišné umístění zpracovávané dávky na jednotlivých strojích. Po ukončení soustružení je přesunuta dávka po objektu lineshape na další stroj, který následuje

v pořadí. Tento pohyb mezi jednotlivými stroji je tvořen problikáváním dané dávky po objektu lineshape, který dané stroje spojuje. Po přesunutí dávky ze soustružení na frézování se oba předešlé stroje vrátí do základní polohy a obrázek dalšího stroje se přepne do pracovní polohy. Následně po ukončení této operace se přesune dávka do skladu, kde setrvá po jistou dobu. Po uplynutí této doby dochází k přesunutí na montážní halu, kde je zvolená zakázka odrer1 ukončena.

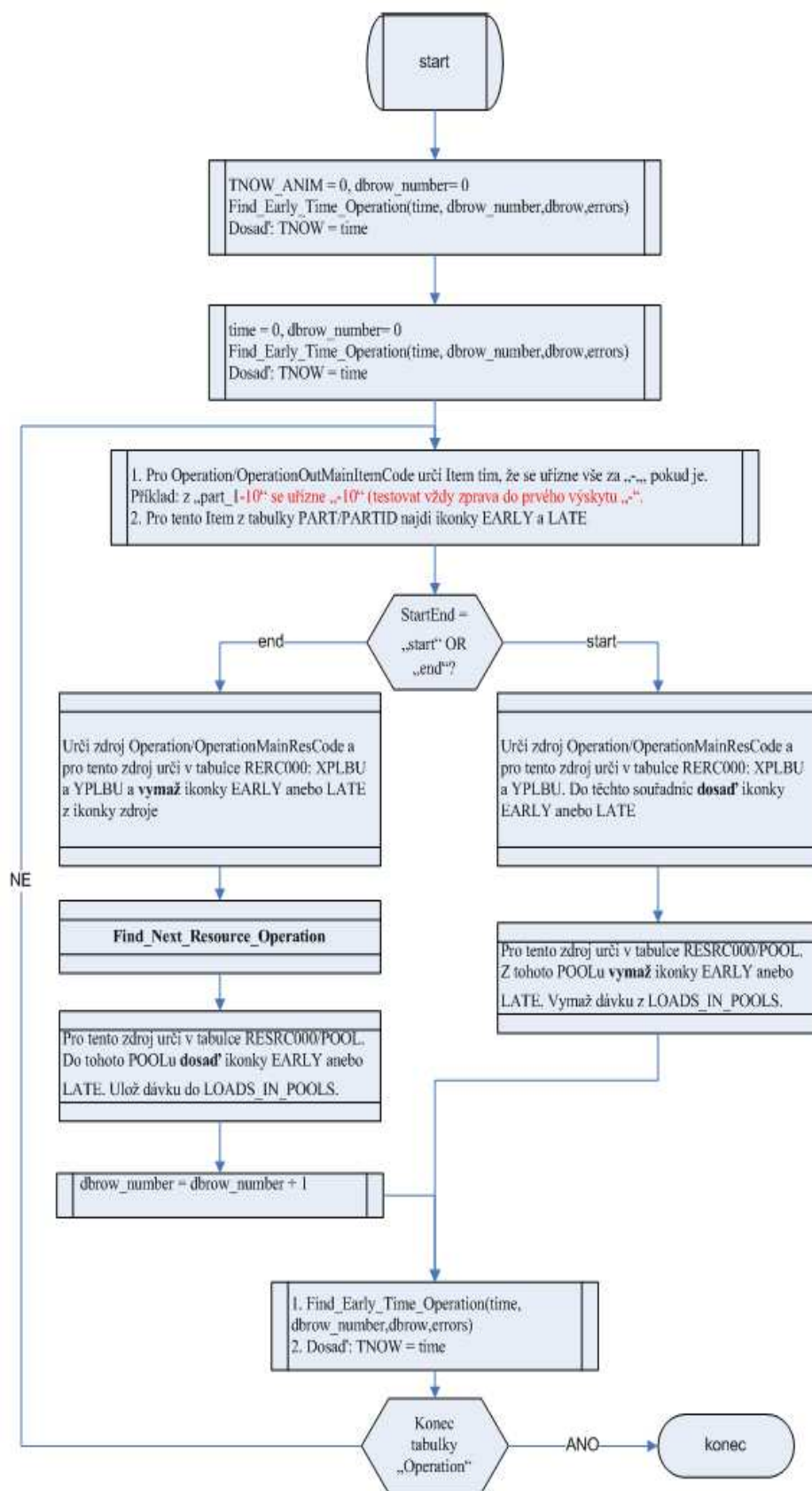
Animace, kterou program zpracovává, je závislá na datech, které se vyskytují ve zkušebním souboru. V daném souboru se nachází tabulka operation. V této tabulce jsou obsaženy veškeré údaje potřebné pro výsledný grafický průběh animace. Jsou zde například časové intervaly, kdy na jednotlivých strojích práce začala nebo skončila. Z této tabulky lze vyčíst propojení mezi navazujícím stroji atd. výsledný program pouze tyto údaje čte, zpracovává a nabízí uživateli možnost nastavitelného grafického zobrazení. Po spuštění animace také dochází k zpřístupnění funkce zastav, které se nachází v hlavním panelu pracovního formuláře. Tato funkce slouží pro zastavení probíhající animace v libovolném kroku. Následně může uživatel použít funkci reset a animaci spustit od začátku. Způsob, kterým program pracuje při vytváření animace, je pro lepší pochopení popsán následujícími blokovými schématy.



Obr.25 Blokové schéma funkce pro nalezení nového zdroje animace



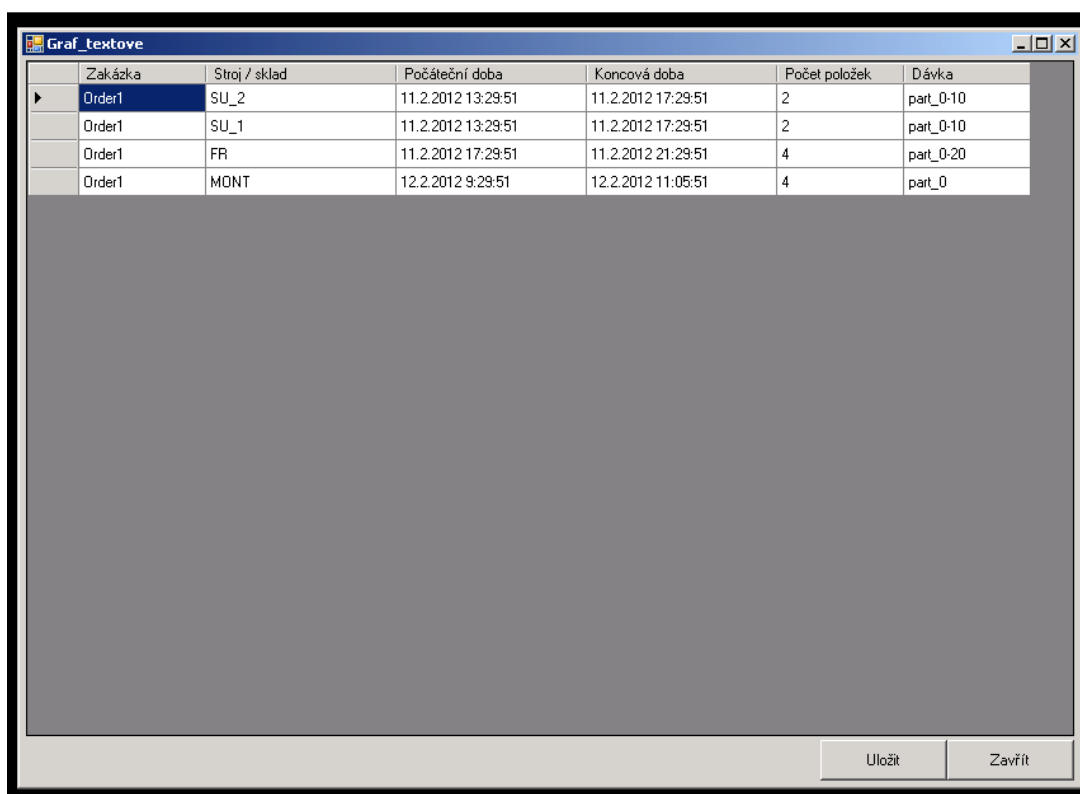
DIPLOMOVÁ PRÁCE



Obr.26 Blokové schéma animace výrobního procesu [14]

2.2. Výstupy simulace výrobního procesu

Po ukončení animace zvoleného výrobního procesu jsou zpřístupněny výstupní data. Tyto výstupní informace se nachází v hlavním panelu pracovního formuláře pod nyní již zpřístupněnou položkou výsledky. Nachází se zde možnost nahlédnutí na textový výpis strojů, které byly využity pro zvolenou zakázku. Daný textový výpis je znázorněn na obr.27.

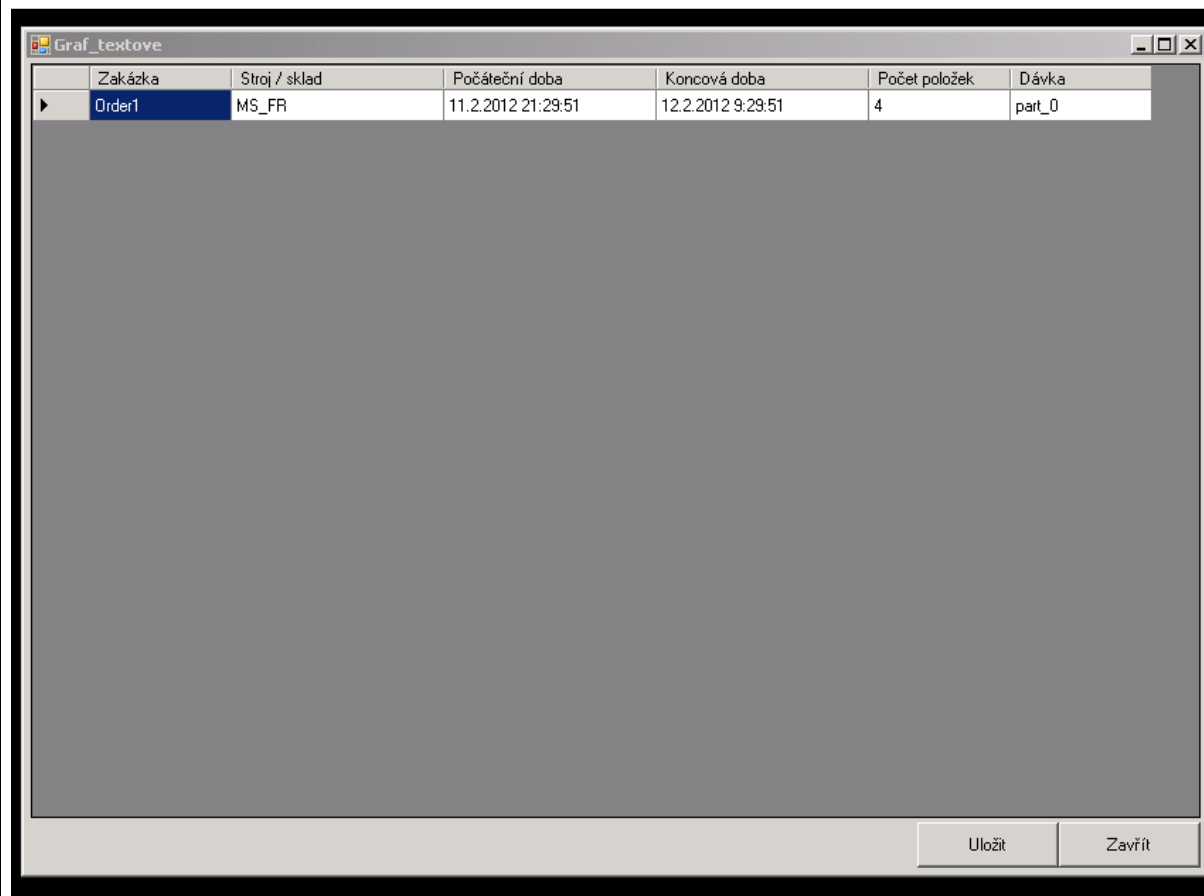


Zakázka	Stroj / sklad	Počáteční doba	Konečná doba	Počet položek	Dávka
Order1	SU_2	11.2.2012 13:29:51	11.2.2012 17:29:51	2	part_0-10
Order1	SU_1	11.2.2012 13:29:51	11.2.2012 17:29:51	2	part_0-10
Order1	FR	11.2.2012 17:29:51	11.2.2012 21:29:51	4	part_0-20
Order1	MONT	12.2.2012 9:29:51	12.2.2012 11:05:51	4	part_0

Obr.27 Textový výstup strojů animované zakázky

Textový výpis strojů je tvořen jednotlivými vybranými položkami ze zkušební souboru, při filtrování pouze vybrané zakázky, která byla použita při animaci. Z obrázku je patrné, kdy práce na jednotlivých strojích začala nebo byla ukončena. Počet přenesených položek a informace o dávce, kterou jednotlivé stroje zpracovávaly. Tento výpis je možno tlačítkem uložit, které vyvolá funkci OpenSaveDialogu, uložit ve formátu txt, csv, xls a xlsx. Předpřipravené šablony pro výstup v Microsoft Excelu však musí být stále součástí programu. Pokud by došlo k jejich poškození nebo smazání, výstup do těchto formátů již nebude možný.

Další z možných výstupů, které se nacházejí pod položkou výsledky, je textový výpis skladů. Tento výpis je znázorněn na obr.28.

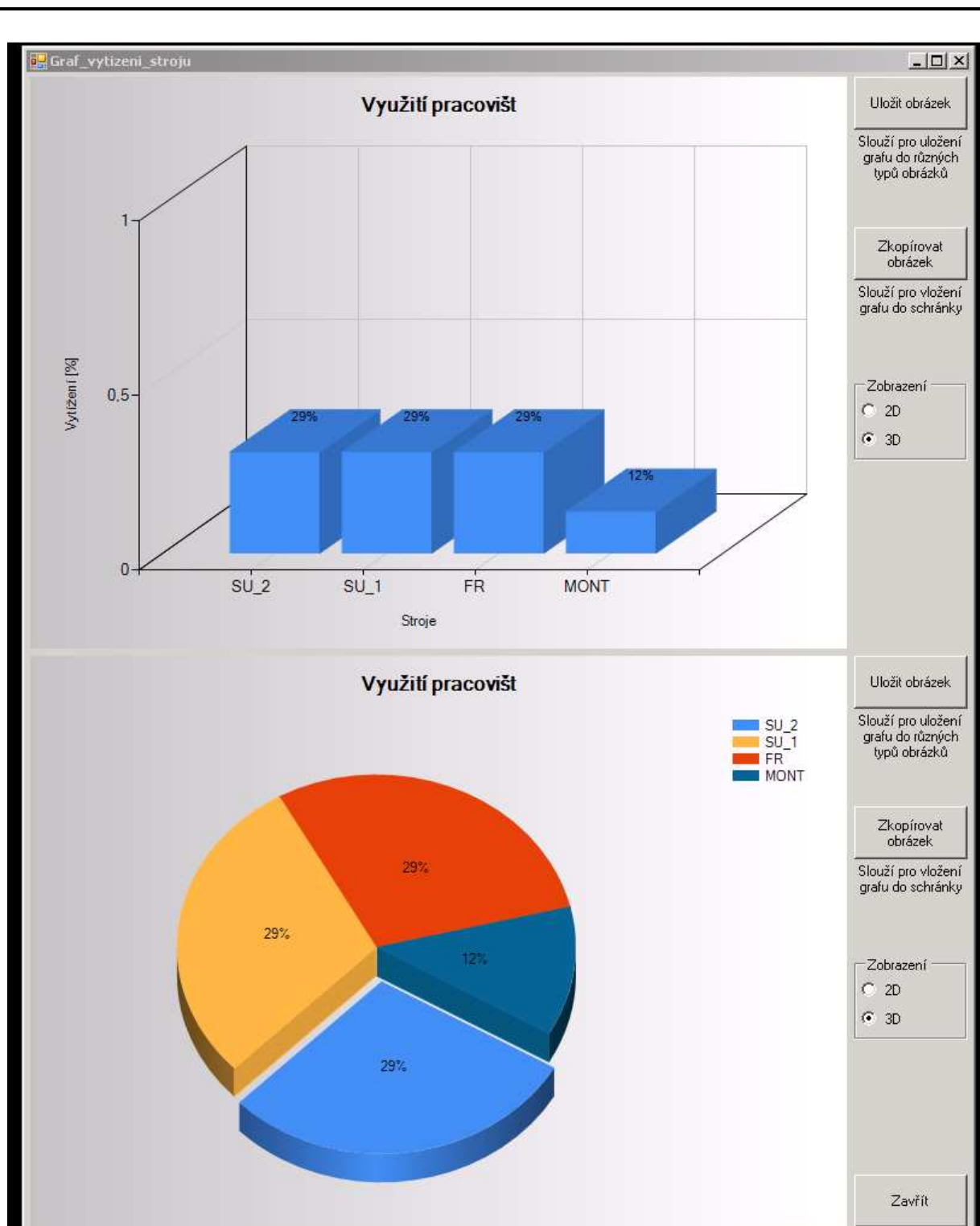
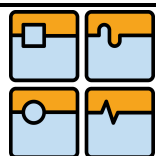


Zakázka	Stroj / sklad	Počáteční doba	Koncová doba	Počet položek	Dávka
Order1	MS_FR	11.2.2012 21:29:51	12.2.2012 9:29:51	4	part_0

Obr.28 Textový výstup skladů animované zakázky

Textový výpis skladů má podobné vlastnosti jako jeho předchůdce. Nachází se zde možnost zjistit, kdy přesně do skladu daná dávka dorazila nebo naopak, kdy sklad opustila. Tento výstup obsahuje i další údaje, které informují například o názvu uložené dávky nebo počty položek, které se v dávce nacházely. Export tohoto výstupu se řídí obdobnými pravidly jako předchozí varianta. Jako poslední možnost textového výstupu se zde nachází možnost exportovat sklady i stroje naráz.

Grafické výstupy, které jsou v této aplikaci k dispozici, byly umožněny za pomoci speciální knihovny DataVizulation.dll. Tato knihovna není součástí standardní verze Visual Studia a musela být proto dodatečně instalována. Knihovna slouží, jako nástroj pro zobrazení jednotlivých grafických výstupů. Prvním z grafických výstupů animace výrobních systému je procentuální využití jednotlivých pracovišť, které se nachází na obr.29



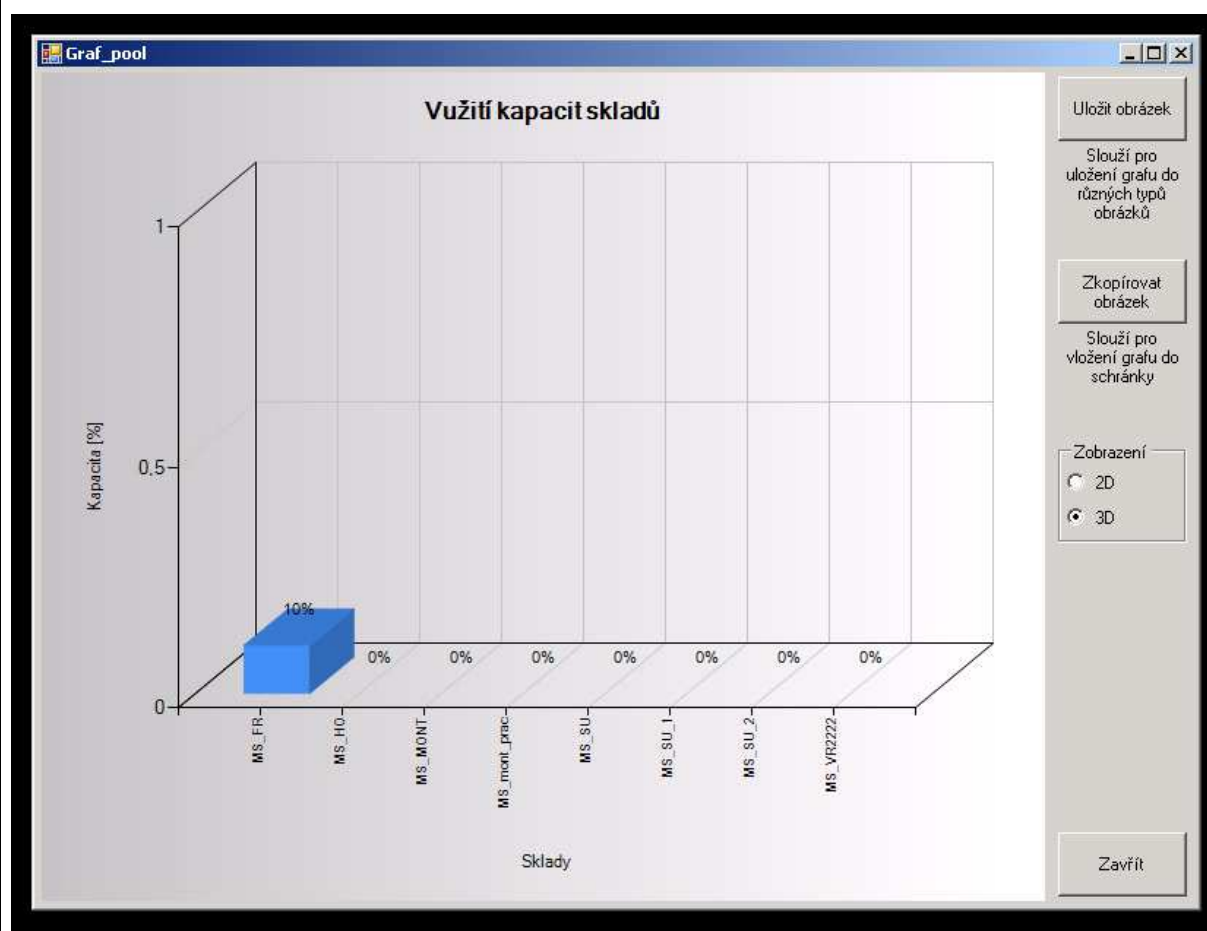
Obr.29 Procentuální vytížení pracovišť pro zakázku Order1

Tento grafický výstup je uživateli k dispozici ve 2D nebo ve 3D variantě. Možnost požadovaného zobrazení je možné přepnout v pravé části grafu. Pro ušetření místa jsem zde proto zvolil pouze 3D zobrazení. Tento výstup má však více vlastností, které uživatel může libovolně použít. Koláčový graf je zde interaktivní a lze

libovolně vysunout jednotlivé stroje. Dále se zde nachází možnost uložení zvolené varianty grafu ve formě obrázku.

Tato funkce je zpřístupněna uživateli při kliknutí na tlačítko uložit obrázek. Vyvolá se SaveFileDialog, který je obdobnou variantou OpenFileDialogu. Využití jednotlivých pracovišť lze potom uložit do množství formátů, které obsahuje filtr. Pro okamžité zkopírování vybrané varianty grafu slouží tlačítko zkopírovat obrázek a následně tento obrázek uživatel může vložit do libovolné aplikace (Microsoft Word, malování) funkcí vložit nebo obdobnou zkratkou.

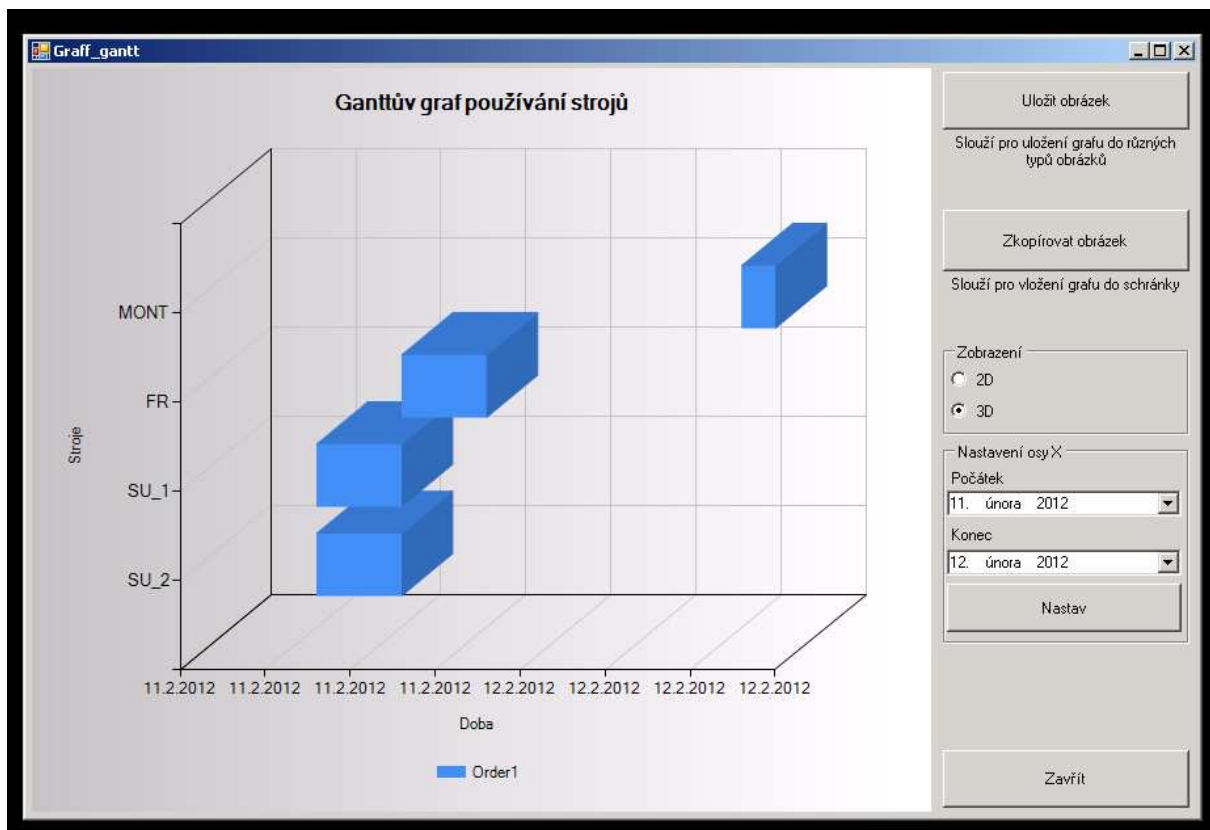
Následujícím obdobným grafickým výstupem je procentuální vytížení skladů. Výstup je tvořen podobným oknem jako jeho předchůdce a nabízí uživateli stejné možnosti práce s tímto grafem. Pro názornost je grafické využití skladů znázorněno na obr.27.



Obr.30 Procentuální využití skladů pro zakázku Order1

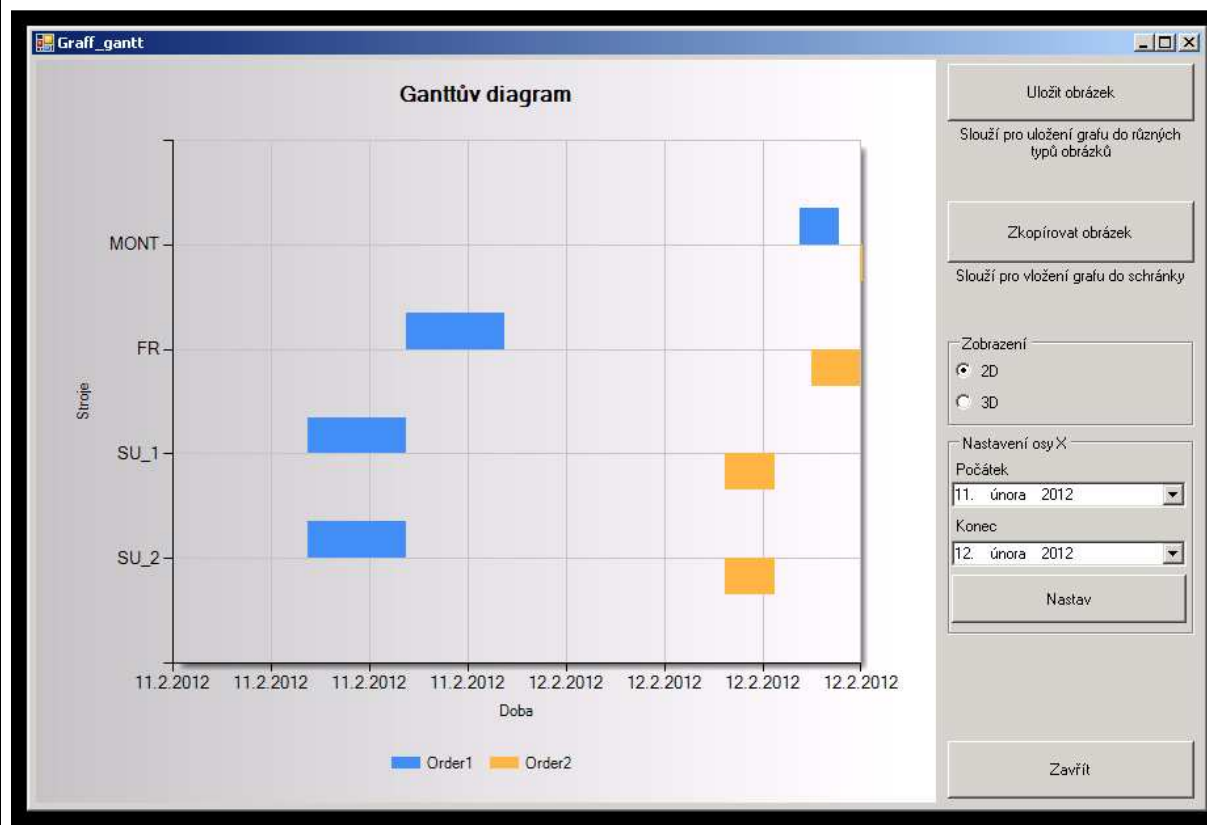
Trochu odlišným grafickým výstupem této aplikace je možnost zobrazení zvolené animace výrobního systému za pomoci Ganttových diagramů. Výstupní graf je opět tvořen stejným formulářem, který uživateli poskytuje stejné funkce pro uchovávání simulovaných výsledků. Okno je oproti ostatním doplněno o možnost

zvolení časového intervalu, v průběhu kterého je možné nahlížet na výrobní proces. Na obr.31 je pro názornost zobrazena zakázka Order1 za pomoci Ganttova diagramu. Na obrázku je zachycena 3D varianta daného diagramu, která je z pohledu uživatele pouze přívětivější. V praxi se ale vyskytují výrobní procesy, které obsahují spousty operací v dlouhodobém časovém intervalu a zde se stává 3D varianta daného diagramu velice nepřehlednou.



Obr.31 Ganttův diagram pro zakázku Order1 ve 3D

Kůly této nepřehlednosti zde proto uvádím také 2D variantu pro simulaci zakázky Order1 a Order2, která je zobrazena na obr.32. Tímto výstupem získá uživatel jistý přehled o tom, jak jednotlivé zakázky výrobního systému byly zpracovávány. Dalším možným výstupem, který tato aplikaci nabízí je diagram zdržení dávek ve frontách, který opět uživateli poskytuje bližší informace o animovaném procesu.



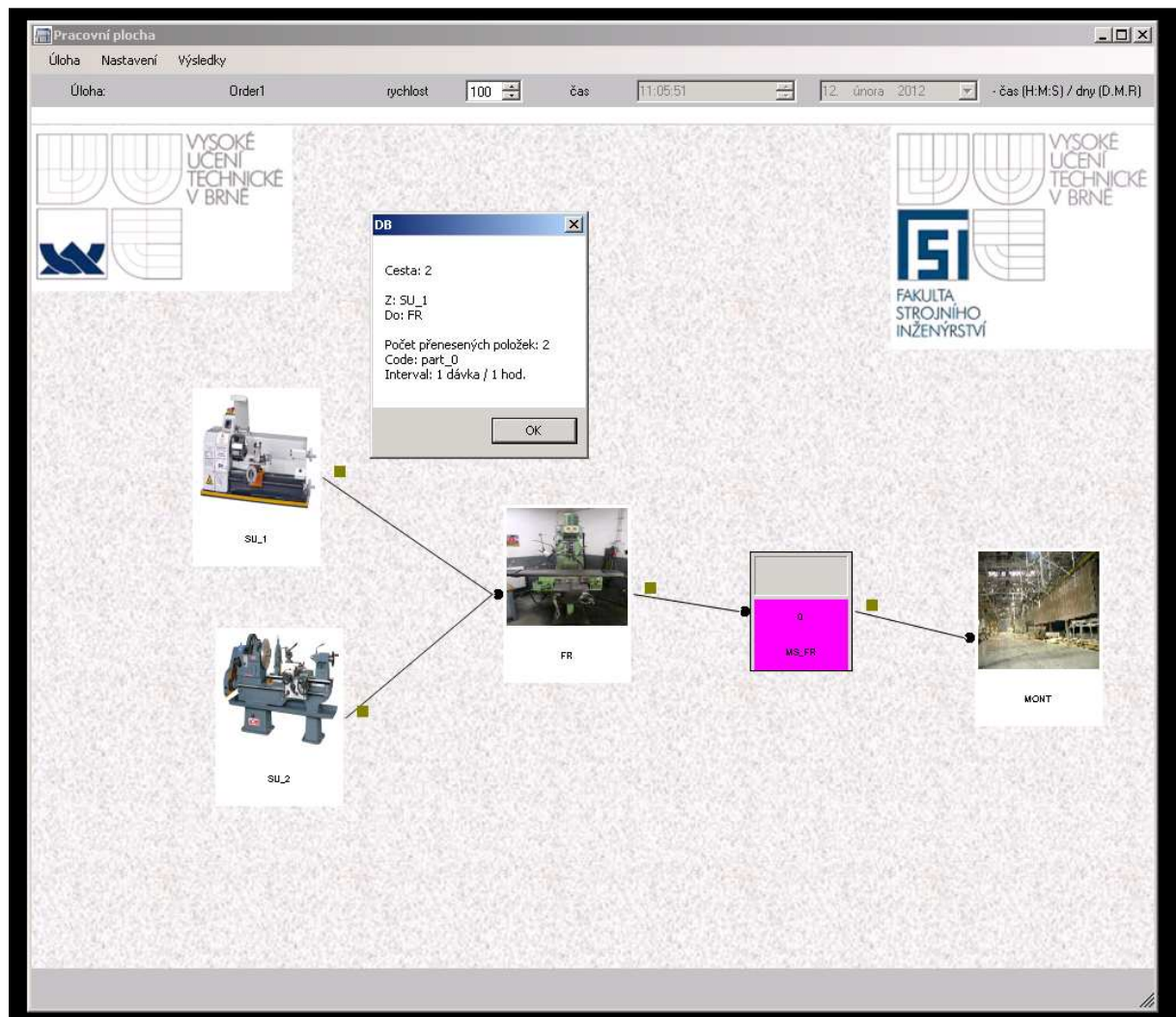
Obr.32 Ganttův diagram pro zakázku Order1 a Order2 ve 2D

Posledním výstupem této praktické části diplomové práce, která uživateli poskytne jistý nadhled nad výrobním procesem, je textový výpis jednotlivých drah, kterými zvolená dávka prochází během animace. Na obr.33 je tento textový výpis zobrazen.

Při vytváření jednotlivých cest mezi stroji se nad každým takovým spojením tvoří hnědý objekt, do kterého jsou zaznamenány informace o průběhu výrobního procesu jednotlivými drahami.

Při kliknutí na daný prvek se zobrazí okno, které uživateli poskytne informace výpisem na obrazovce. Uživatel má tak okamžitou informaci o tom, co po prvku lineshape probíhalo například za dávku. Je zde možnost snadno dohledat množství přenesených prvků z jednoho stroje na další. S množstvím přenesených prvků zvětšuje objekt lineshape svoji šířku.

Jelikož jsme v tomto případě animovali velmi drobnou zakázku, není tato změna spojovacích drah jednotlivých pracovišť patrná. Pro uživatele je v tomto případě určující pouze textový výpis dané cesty.



Obr.33 Textový výpis jednotlivých cest mezi prvky systému.

Zhodnocení

V diplomové práci je zpracován přehled vývoje simulačního software. Jsou zde rozepsány výhody a nevýhody, které využití tohoto softwaru přináší. Dále je práce podrobněji zaměřena na využití a realizaci animací. V této části práce jsou uvedeny základy animace. Dále je zde podrobněji popsána realizace počítačové animace a jsou zde vysvětleny jednotlivé pojmy, které s počítačovou animací souvisejí. Teoretická část diplomové práce je zakončena přiblížením technologii a software, který byl využit při zhotovení praktické části práce.

Praktická část diplomové práce je zaměřena na popis a funkcí programového rozhraní, které je výstupem této práce. Hlavním požadavkem vedoucího této diplomové práce bylo vytvořit program, který by byl schopen animovat výrobní proces za pomoci předložených dat. Nejedlo se pouze o program na jedno použití, ale o univerzální nástroj, který by byl schopen animovat i jiná vstupní data v podobném formátu. Tento požadavek se v diplomové práci podařilo splnit. Další z cílů diplomové práce byla možnost uživatele vlastního grafického nastavení pracovní plochy animace. Uživatel této aplikace je schopen nastavit si jakékoliv pozadí pro svoji animaci ve formě barvy nebo obrázku. Dále se v programu nachází také možnost, měnit obrázky objektů, které reprezentují stroje či sklady. Uživatel si také může dle potřeby nastavit i polohu a barvu objektu, který reprezentuje zpracovávanou dávku výrobního procesu. V praktické části diplomové práce je velice zjednodušeně popsán postup, kterým výsledný program data zpracovává a realizuje celou animaci zvolené výrobní zakázky. Pro získání více informací o dané funkci a realizaci celého projektu stačí nahlédnout na přiložené CD, kde se nachází okomentovaných zdrojový kód programu. Závěrečná část diplomové práce je věnovaná výstupům, které program uživateli nabízí. Uživatel má možnost získat textové informace o animovaném procesu. Dále má uživatel možnost nahlédnout na procentuální vytížení skladu či strojů zvolené animované zakázky, získat přehled o jednotlivých dávkách čekající ve frontě. Nachází se zde také možnost získat lepší náhled nad výrobním systémem za pomoci Ganntových či Sankey diagramů.

Výsledný program diplomové práce zajisté nedosahuje úrovně, na kterých pracují profesionální programy, které jsou pro tyto účely animace vyrobeny, ale poskytuje uživateli alespoň příjemné modifikovatelné grafické prostředí, ve kterém si může animaci zobrazit a získat tak alespoň základní představu o tom, jakým způsobem výrobní proces funguje.

Seznam použitých zdrojů

- [1] Doc.Ing. Václav Votava, CSc., Ing. Zdeněk Ulrych, Ph.D., Ing. Pavel Raška
Ing. Petr Hořejší: *Simulace ve strojírenství*, ebook verze 2., Západočeská
univerzita Plzeň, 2008.
- [2] MANLIG, František. HUMUSOFT. [online]. [cit. 2012-06-30]. Dostupné z:
<http://www.humusoft.cz/archived/pub/witness/9910/manlig.htm>
- [3] SKOUPIL, Martin. DesignTech: Tecnomatix Plant Simulation. [online]. [cit.
2012-06-30]. Dostupné z: <http://www.designtech.cz/c/cad/plm/plm/tecnomatix-plant-simulation.htm>
- [4] KLČOVÁ, Hana, Petr SODOMKA a Jaromír HABAN. CVIS: Na český trh
přichází originální tuzemské APS řešení. [online]. [cit. 2012-06-30]. Dostupné z:
<http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=488>
- [5] Wikipedie: Animace. [online]. [cit. 2012-06-30]. Dostupné z:
<http://cs.wikipedia.org/wiki/Animace>
- [6] BÍZKOVÁ, Milada. Počítačová grafika. [online]. [cit. 2012-06-30]. Dostupné z:
<http://home.zcu.cz/~bizkova/>
- [7] . ENotes: .NET Framework. [online]. [cit. 2012-06-30]. Dostupné z:
http://www.enotes.com/topic/.NET_Framework
- [8] SWMAG: Architektura .NET Framework rozebrána do podrobnosti. [online]. [cit.
2012-06-30]. Dostupné z: <http://www.swmag.cz/670/architektura-net-framework-rozebrana-do-podrobna/>
- [9] ALTINDAG, Huseyin. THE Code Project: Using ADO.NET for beginners.
[online]. [cit. 2012-06-30]. Dostupné z:
<http://www.codeproject.com/Articles/8477/Using-ADO-NET-for-beginners>
- [10] E-learningové prvky pro podporu výuky odborných a technických předmětů:
Vše o animaci. [online]. [cit. 2012-06-30]. Dostupné z:
<http://www.elearn.vsb.cz/archivcd/FEI/ZPG/00/9.pdf>
- [11] Wikipedia: Počítačová animace. [online]. [cit. 2012-06-30]. Dostupné z:
http://en.wikipedia.org.cs.mk.gd/wiki/Computer_animation
- [12] Metalcity: Počítačová animace. [online]. [cit. 2012-06-30]. Dostupné z:
<http://www.metalcity.estranky.cz/clanky/pocitacova-animace.html>

Seznam použitých obrázků

Obr.1 Princip simulace	10
Obr.2 Příklad použití Sankey diagramu v programu Plant Simulation	15
Obr.3 Příklad použití Ganttova diagramu v aplikaci ASP Karat	16
Obr. 4. Příklad průběhu animace	16
Obr. 5. Příklad vytvoření animace v programu Motion Capture	18
Obr. 6 Převod obrazu do bitmapové grafiky	20
Obr. 7 Ukázka efektivity vektorové grafiky při zvětšování	21
Obr. 8 Bézierova křivka	22
Obr. 9 Pracovní prostředí Microsoft Visual Studio 2010 Express.....	24
Obr. 10 Schéma struktury .NET Framework	25
Obr.11 Vývoj .NET Frameworku v průběhu let	26
Obr.12 Objektový model ADO .NET	27
Obr.13 Úvodní okno animačního programu	28
Obr.14 Seznam základních možností programu	29
Obr.15 Vyvolání OpenFileDialog pro vyhledávání souboru Microsoft Access	30
Obr.16 Úvodní okno aplikace po načtení zkušebního souboru	31
Obr.17 Náhled do zkušebního souboru pro animaci výrobních procesů	32
Obr.18 Okno pracovního formuláře pro zakázku Order1.....	33
Obr.19 Formulář, který umožňuje nastavení pracovní plochy	34
Obr.20 Vzorek barvy, kterou je možno nastavit jako pozadí pracovní plochy.	35
Obr.21 Okno, které slouží pro nastavení vlastností stroje.....	36
Obr.22 Formulář, který slouží pro nastavení obléžků stroje	36
Obr.23 Formulář, který slouží pro nastavení skladu	37
Obr.24 Okno pracovní plochy při spuštění animace.....	38
Obr.25 Blokové schéma funkce pro nalezení nového zdroje animace	39
Obr.26 Blokové schéma animace výrobního procesu	40
Obr.27 Textový výstup strojů animované zakázky	41
Obr.28 Textový výstup skladů animované zakázky.....	42
Obr.29 Procentuální vytížení pracovišť pro zakázku Order1	43
Obr.30 Procentuální využití skladů.....	44
Obr.31 Ganttův diagram pro zakázku Order1 ve 3D	45
Obr.32 Ganttův diagram pro zakázku Order1 a Order2 ve 2D	46
Obr.33 Textový výpis jednotlivých cest mezi prvky systému.....	47

Seznam použitých tabulek

Tab. 1 Formáty bitmapové grafiky	20
Tab. 2 Formáty vektorové grafiky.....	22

Seznam použitých zkratk a symbolů

ADO	ActiveX Data Objects
ADO.NET	ActiveX Data Objects .NET
BCL	Base Class Library
CAD	Computer-aided design
CLR	Common Language Runtime
CRT	Cathode ray tube
GASP	General Activity Simulation Program
GASP	General Activity Simulation Program
GPSS	General Purpose Simulation Systém
IBM	nternational Business Machines Corporation
MSIL	Microsoft Intermediate Language
OOP	Object Oriented Programming
RGB	red, green, blue
SDX	Simulation Data Exchange
SGI	Silicon Graphics Inc
SIGGRAPH	Special Interest Group on GRAPHics and Interactive Techniques
SIMAN	SIMulation ANalysis
SIMULA	SIMUlation LAnguage)
SLAM	Simulation Language for Alternative Modeling
XML	Extensible Markup Language

Přílohy

Příložené CD obsahuje tyto základní složky:

- *bin* – spustitelná demonstrační aplikace pro Windows
- *test* – zkušební soubor pro testování programu
- *pdf* – diplomová práce v PDF formátu
- *sources* - zdrojové soubory aplikace (aplikace ADO.NET je programovaná v jazyce VisualBasic .NET)